

# Package: tbeptools (via r-universe)

October 10, 2024

**Title** Data and Indicators for the Tampa Bay Estuary Program

**Version** 3.0.0.9000

**Date** 2024-10-10

**Description** Several functions are provided for working with Tampa Bay Estuary Program data and indicators, including the water quality report card, tidal creek assessments, Tampa Bay Nekton Index, Tampa Bay Benthic Index, seagrass transect data, habitat report card, and fecal indicator bacteria. Additional functions are provided for miscellaneous tasks, such as reference library curation.

**Depends** R (>= 3.6), methods

**Imports** dataRetrieval, dplyr, flextable, ggplot2 (>= 3.2.0), httr, jsonlite, leaflet, lubridate, patchwork, plotly, purrr, reactable, readxl, rnoaa, sf, tibble, tidyr, tools

**Suggests** bookdown, ftExtra, ggmap, ggnewscale, ggrepel, ggspatial, htmlwidgets, knitr, mapview, mockery, rmarkdown, testthat (>= 2.1.0), covr

**Remotes** ropensci/rnoaa

**License** MIT + file LICENSE

**Encoding** UTF-8

**VignetteBuilder** knitr

**LazyData** true

**LazyDataCompression** xz

**RoxygenNote** 7.2.3

**URL** <https://github.com/tbep-tech/tbeptools>

**BugReports** <https://github.com/tbep-tech/tbeptools/issues>

**Repository** <https://tbep-tech.r-universe.dev>

**RemoteUrl** <https://github.com/tbep-tech/tbeptools>

**RemoteRef** HEAD

**RemoteSha** 8d5efcd84207fd902a573d2066f90001eda221c4

## Contents

acres . . . . .	4
anlz_attain . . . . .	5
anlz_attainsite . . . . .	6
anlz_avedat . . . . .	7
anlz_avedatsite . . . . .	7
anlz_enteromap . . . . .	8
anlz_fibmap . . . . .	9
anlz_fibmatrix . . . . .	11
anlz_fibwetdry . . . . .	13
anlz_hmpreport . . . . .	14
anlz_hydroload . . . . .	16
anlz_iwrraw . . . . .	17
anlz_refs . . . . .	17
anlz_sedimentaddtot . . . . .	18
anlz_sedimentave . . . . .	19
anlz_sedimentpel . . . . .	20
anlz_sedimentpelave . . . . .	21
anlz_tbbimed . . . . .	22
anlz_tbbiscr . . . . .	23
anlz_tbniave . . . . .	23
anlz_tbnimet . . . . .	24
anlz_tbniscr . . . . .	25
anlz_tdlcrk . . . . .	26
anlz_tdlcrkindic . . . . .	26
anlz_transectave . . . . .	27
anlz_transectavespp . . . . .	28
anlz_transectocc . . . . .	30
anlz_yrattain . . . . .	31
benthicdata . . . . .	31
bsmap . . . . .	32
catchpixels . . . . .	32
catchprecip . . . . .	33
enterodata . . . . .	34
epcdata . . . . .	35
fibdata . . . . .	36
fimdata . . . . .	37
fimstations . . . . .	38
hmptrgs . . . . .	39
iwrraw . . . . .	39
phytodata . . . . .	40
read_dlcurrent . . . . .	41
read_formbenthic . . . . .	42
read_formfib . . . . .	43
read_formfim . . . . .	44
read_formphyto . . . . .	45
read_formsediment . . . . .	46

read_formtransect . . . . .	47
read_formwq . . . . .	48
read_formwqp . . . . .	49
read_importbenthic . . . . .	50
read_importentero . . . . .	51
read_importepc . . . . .	52
read_importfib . . . . .	53
read_importfim . . . . .	55
read_importphyto . . . . .	56
read_importrain . . . . .	57
read_importsediment . . . . .	57
read_importwq . . . . .	58
read_importwqp . . . . .	60
read_transect . . . . .	61
seagrass . . . . .	62
sedimentdata . . . . .	63
sgmanagement . . . . .	64
sgseg . . . . .	65
show_annualassess . . . . .	66
show_boxplot . . . . .	67
show_complot . . . . .	68
show_enteromap . . . . .	69
show_fibmap . . . . .	71
show_fibmatmap . . . . .	72
show_fibmatrix . . . . .	74
show_hmpreport . . . . .	76
show_matrix . . . . .	78
show_matrixplotly . . . . .	79
show_ratab . . . . .	80
show_reactable . . . . .	81
show_seagrasscoverage . . . . .	82
show_sedimentalratio . . . . .	83
show_sedimentave . . . . .	85
show_sedimentmap . . . . .	86
show_sedimentpelave . . . . .	87
show_sedimentpelaveplotly . . . . .	88
show_sedimentpelmap . . . . .	89
show_segmatrix . . . . .	90
show_segplotly . . . . .	91
show_sitemap . . . . .	92
show_sitesegmap . . . . .	93
show_tbbimatrix . . . . .	94
show_tbnimatrix . . . . .	95
show_tbniscr . . . . .	96
show_tbniscrall . . . . .	97
show_tbniscrplotly . . . . .	98
show_tdlcrk . . . . .	99
show_tdlcrkindic . . . . .	99

show_tdlcrkindiccdf . . . . .	100
show_tdlcrkline . . . . .	101
show_tdlcrkmatrix . . . . .	102
show_tdlcrkradar . . . . .	103
show_thrplot . . . . .	104
show_transect . . . . .	105
show_transectavespp . . . . .	106
show_transectmatrix . . . . .	108
show_transectsum . . . . .	110
show_wqmatrix . . . . .	111
stations . . . . .	112
subt acres . . . . .	113
swfwmdtbseg . . . . .	113
targets . . . . .	114
tbniref . . . . .	115
tbnispp . . . . .	116
tbseg . . . . .	117
tbsegdetail . . . . .	118
tbseglines . . . . .	118
tbsegshed . . . . .	119
tbshed . . . . .	119
tidalcreeks . . . . .	120
tidaltargets . . . . .	121
transect . . . . .	122
trnlns . . . . .	123
trnpts . . . . .	123
util_fibicons . . . . .	124
util_fiblevs . . . . .	125
util_html . . . . .	125
util_map . . . . .	126
util_organ . . . . .	127

**Index****128**

acres

*Tampa Bay intertidal and supratidal land use and cover***Description**

Tampa Bay intertidal and supratidal land use and cover for Habitat Master Plan reporting

**Usage**

acres

**Format**

A data frame

## Examples

```
## Not run:  
load(url("https://github.com/tbep-tech/hmpu-workflow/raw/master/data/acres.RData"))  
  
save(acres, file = 'data/acres.RData', compress = 'xz')  
  
## End(Not run)
```

---

anlz\_attain

*Get attainment categories*

---

## Description

Get attainment categories for each year and bay segment using chlorophyll and light attenuation

## Usage

```
anlz_attain(avedat, magdurout = FALSE, trgs = NULL)
```

## Arguments

avedat	result returned from <a href="#">anlz_avedat</a>
magdurout	logical indicating if the separate magnitude and duration estimates are returned
trgs	optional data.frame for annual bay segment water quality targets, defaults to <a href="#">targets</a>

## Value

A data.frame for each year and bay segment showing the attainment category

## Examples

```
avedat <- anlz_avedat(epcdata)  
anlz_attain(avedat)
```

---

anz\_attainsite      *Get site attainments*

---

### Description

Get site attainment categories for chlorophyll or light attenuation

### Usage

```
anz_attainsite(  
  avedatsite,  
  thr = c("chl", "la"),  
  trgs = NULL,  
  yrrng = NULL,  
  thrs = FALSE  
)
```

### Arguments

avedatsite	result returned from <a href="#">anz_avedatsite</a>
thr	chr string indicating with water quality value and appropriate threshold to plot, one of "chl" for chlorophyll and "la" for light availability
trgs	optional data.frame for annual bay segment water quality targets, defaults to <a href="#">targets</a>
yrrng	optional numeric value for year to return, defaults to all
thrs	logical indicating if attainment category is relative to targets (default) or thresholds

### Details

This function is a simplification of the attainment categories returned by [anz\\_attain](#). Sites are only compared to the targets/thresholds that apply separately for chlorophyll or light attenuation.

### Value

a data.frame for each year and site showing the attainment category

### Examples

```
avedatsite <- anz_avedatsite(epcdata)  
anz_attainsite(avedatsite)
```

---

anlz_avedat	<i>Estimate annual means</i>
-------------	------------------------------

---

**Description**

Estimate annual means for chlorophyll and secchi data

**Usage**

```
anlz_avedat(epcdata, partialyr = FALSE)
```

**Arguments**

epcdata	data.frame formatted from read_importtwq
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

**Value**

Mean estimates for chlorophyll and secchi

**Examples**

```
# view average estimates  
anlz_avedat(epcdata)
```

---

anlz_avedatsite	<i>Estimate annual means by site</i>
-----------------	--------------------------------------

---

**Description**

Estimate annual means by site for chlorophyll and secchi data

**Usage**

```
anlz_avedatsite(epcdata, partialyr = FALSE)
```

**Arguments**

epcdata	data.frame formatted from read_importtwq
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

**Value**

Mean estimates for chlorophyll and secchi

**Examples**

```
# view average estimates
anlz_avedatsite(epcdata)
```

---

```
anlz_enteromap      Assign threshold categories to Enterococcus data
```

---

**Description**

Assign threshold categories to Enterococcus data

**Usage**

```
anlz_enteromap(
  fibdata,
  yrsel = NULL,
  mosel = NULL,
  areasel = NULL,
  wetdry = FALSE,
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  assf = FALSE
)
```

**Arguments**

fibdata	data frame of Enterococcus sample data as returned by <a href="#">enterodata</a> or <a href="#">anlz_fibwetdry</a>
yrsel	optional numeric to filter data by year
mosel	optional numeric to filter data by month
areasel	optional character string to filter output by stations in the long_name column of enterodata, see details
wetdry	logical; if TRUE, incorporate wet/dry differences (this will result in a call to <a href="#">anlz_fibwetdry</a> , in which case temporal_window and wet_threshold are required). If FALSE (default), do not differentiate between wet and dry samples.
precipdata	input data frame as returned by <a href="#">read_importrain</a> . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object <a href="#">catchprecip</a> has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to <a href="#">catchprecip</a> .
temporal_window	numeric; required if wetdry is TRUE. number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric; required if wetdry is TRUE. inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
assf	logical indicating if the data are further processed as a simple features object with additional columns for <a href="#">show_enteromap</a>



## Details

This function is based on [anlz\\_fibmap](#), but is specific to Enterococcus data downloaded via [read\\_importentero](#). It creates categories for mapping using [show\\_enteromap](#). Optionally, if samples have been defined as 'wet' or not via [anlz\\_fibwetdry](#), this can be represented via symbols on the map. Categories based on relevant thresholds are assigned to each observation. The categories are specific to Enterococcus in marine waters (class of 2 or 3M). A station is categorized into one of four ranges defined by the thresholds as noted in the `cat` column of the output, with corresponding colors appropriate for each range as noted in the `col` column of the output.

The `areasel` argument can indicate valid entries in the `long_name` column of `enterodata`. For example, use "Old Tampa Bay" for stations in the subwatershed of Old Tampa Bay, where rows in `enterodata` are filtered based on the the selection. All stations are returned if this argument is set as NULL (default). All valid options for `areasel` include "Old Tampa Bay", "Hillsborough Bay", "Middle Tampa Bay", "Lower Tampa Bay", "Boca Ciega Bay", or "Manatee River". One to any of the options can be used.

## Value

A data.frame similar to `fibdata` if `assf = FALSE` with additional columns describing station categories and optionally filtered by arguments passed to the function. A `sf` object if `assf = TRUE` with additional columns for [show\\_enteromap](#).

## Examples

```
anlz_enteromap(enterodata, yrsel = 2020, mosel = 9)

# differentiate wet/dry samples in that time frame
anlz_enteromap(enterodata, yrsel = 2020, mosel = 9, wetdry = TRUE,
               temporal_window = 2, wet_threshold = 0.5)

# as sf object
anlz_enteromap(enterodata, assf = TRUE)
```

---

`anlz_fibmap`

*Assign threshold categories to Fecal Indicator Bacteria (FIB) data*

---

## Description

Assign threshold categories to Fecal Indicator Bacteria (FIB) data

## Usage

```
anlz_fibmap(fibdata, yrsel = NULL, mosel = NULL, areasel = NULL, assf = FALSE)
```

## Arguments

fibdata	input FIB data.frame as returned by <a href="#">read_importfib</a>
yrsel	optional numeric value to filter output by years in fibdata
mosel	optional numeric value to filter output by month in fibdata
areasel	optional character string to filter output by stations in the area column of fibdata, see details
assf	logical indicating if the data are further processed as a simple features object with additional columns for <a href="#">show_fibmap</a>

## Details

This function is used to create FIB categories for mapping using [show\\_fibmap](#). Categories based on relevant thresholds are assigned to each observation. The categories are specific to E. coli or Enterococcus and are assigned based on the station class as freshwater (class as 1 or 3F) or marine (class as 2 or 3M), respectively. A station is categorized into one of four ranges defined by the thresholds as noted in the cat column of the output, with corresponding colors appropriate for each range as noted in the col column of the output.

The areasel argument can indicate valid entries in the area column of fibdata. For example, use either "Alafia River" or "Hillsborough River" for the corresponding river basins, where rows in fibdata are filtered based on the the selection. All stations are returned if this argument is set as NULL (default). The Alafia River basin includes values in the area column of fibdata as "Alafia River" and "Alafia River Tributary". The Hillsborough River basin includes values in the area column of fibdat as "Hillsborough River", "Hillsborough River Tributary", "Lake Thonotosassa", "Lake Thonotosassa Tributary", and "Lake Roberta". Not all areas may be present based on the selection. All valid options for areasel include "Alafia River", "Hillsborough River", "Big Bend", "Cockroach Bay", "East Lake Outfall", "Hillsborough Bay", "Little Manatee", "Lower Tampa Bay", "McKay Bay", "Middle Tampa Bay", "Old Tampa Bay", "Palm River", "Tampa Bypass Canal", or "Valrico Lake". One to any of the options can be used.

## Value

A data.frame if similar to fibdata if assf = FALSE with additional columns describing station categories and optionally filtered by arguments passed to the function. A sf object if assf = TRUE with additional columns for [show\\_fibmap](#).

## Examples

```
# assign categories to all
anlz_fibmap(fibdata)

# filter by year, month, and area
anlz_fibmap(fibdata, yrsel = 2020, mosel = 7, areasel = 'Alafia River')

# as sf object
anlz_fibmap(fibdata, assf = TRUE)
```

---

anlz_fibmatrix	<i>Analyze Fecal Indicator Bacteria categories over time by station or bay segment</i>
----------------	--

---

## Description

Analyze Fecal Indicator Bacteria categories over time by station or bay segment

## Usage

```
anlz_fibmatrix(
  fibdata,
  yrrng = NULL,
  stas = NULL,
  bay_segment = NULL,
  indic,
  threshold = NULL,
  lagyr = 3,
  subset_wetdry = c("all", "wet", "dry"),
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  warn = TRUE
)
```

## Arguments

fibdata	input data frame as returned by <a href="#">read_importfib</a> or <a href="#">read_importentero</a>
yrrng	numeric vector indicating min, max years to include, defaults to range of years in data, see details
stas	optional vector of stations to include, see details
bay_segment	optional vector of bay segment names to include, supercedes stas if provided, see details
indic	character for choice of fecal indicator. Allowable options are <code>fcolif</code> for fecal coliform, or <code>entero</code> for Enterococcus. A numeric column in the data frame must have this name.
threshold	optional numeric for threshold against which to calculate exceedances for the indicator bacteria of choice. If not provided, defaults to 400 for <code>fcolif</code> and 130 for <code>entero</code> .
lagyr	numeric for year lag to calculate categories, see details
subset_wetdry	character, subset data frame to only wet or dry samples as defined by <code>wet_threshold</code> and <code>temporal_window</code> ? Defaults to "all", which will not subset. If "wet" or "dry" is specified, <a href="#">anlz_fibwetdry</a> is called using the further specified parameters, and the data frame is subsetted accordingly.

precipdata	input data frame as returned by <code>read_importtrain</code> . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object <code>catchprecip</code> has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to <code>catchprecip</code> .
temporal_window	numeric; required if <code>subset_wetdry</code> is not "all". number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric; required if <code>subset_wetdry</code> is not "all". inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
warn	logical to print warnings about stations with insufficient data, default TRUE

### Details

This function is used to create output for plotting a matrix stoplight graphic for FIB categories by station. The output can also be summarized by bay segment if `bay_segment` is not NULL and the input data is from `read_importentero`. In the latter case, the `stas` argument is ignored and all stations within each subsegment watershed are used to evaluate the FIB categories. Each station (or bay segment) and year combination is categorized based on the likelihood of fecal indicator bacteria concentrations exceeding some threshold in a given year. For fecal coliform, the default threshold is 400 CFU / 100 mL in a given year (using Fecal Coliform, `fcolif` in `fibdata`). For Enterococcus, the default threshold is 130 CFU / 100 mL. The proportions are categorized as A, B, C, D, or E (Microbial Water Quality Assessment or MWQA categories) with corresponding colors, where the breakpoints for each category are <10%, 10-30%, 30-50%, 50-75%, and >75% (right-closed). By default, the results for each year are based on a right-centered window that uses the previous two years and the current year to calculate probabilities using the monthly samples (`lagyr` = 3). See `show_fibmatrix` for additional details.

`yrrng` can be specified several ways. If `yrrng` = NULL, the year range of the data for the selected changes is chosen. User-defined values for the minimum and maximum years can also be used, or only a minimum or maximum can be specified, e.g., `yrrng = c(2000, 2010)` or `yrrng = c(2000, NA)`. In the latter case, the maximum year will be defined by the data.

The default stations for fecal coliform data are those used in TBEP report #05-13 (<https://drive.google.com/file/d/1MZnK3cMzV7LRg6dTbCKX8A0ZU0GNurJJ/view>) for the Hillsborough River Basin Management Action Plan (BMAP) subbasins if `bay_segment` is NULL and the input data are from `read_importfib`. These include Blackwater Creek (WBID 1482, EPC stations 143, 108), Baker Creek (WBID 1522C, EPC station 107), Lake Thonotosassa (WBID 1522B, EPC stations 135, 118), Flint Creek (WBID 1522A, EPC station 148), and the Lower Hillsborough River (WBID 1443E, EPC stations 105, 152, 137). Other stations can be plotted using the `stas` argument.

### Value

A `tibble` object with FIB summaries by year and station including columns for the estimated geometric mean of fecal coliform or Enterococcus concentrations (`gmean`), the proportion of samples exceeding 400 CFU / 100 mL (fecal coliform) or 130 CFU / 100 mL (Enterococcus) (`exceed`), the count of samples (`cnt`), and a category indicating a letter outcome based on the proportion of exceedences (`cat`). Results can be summarized by bay segment if `bay_segment` is not NULL and the input data is from `read_importentero`.

**See Also**[show\\_fibmatrix](#)**Examples**

```

anlz_fibmatrix(fibdata, indic = 'fcolif')

# use different indicator
anlz_fibmatrix(fibdata, indic = 'entero')

# use different dataset
anlz_fibmatrix(enterodata, indic = 'entero', lagyr = 1)

# same entero data; lower threshold - changes 'cat' scores
anlz_fibmatrix(enterodata, indic = 'entero', lagyr = 1, threshold = 30)

# subset to only wet samples
anlz_fibmatrix(enterodata, indic = 'entero', lagyr = 1, subset_wetdry = "wet",
               temporal_window = 2, wet_threshold = 0.5)

# subset to only dry samples
anlz_fibmatrix(enterodata, indic = 'entero', lagyr = 1, subset_wetdry = "dry",
               temporal_window = 2, wet_threshold = 0.5)

```

---

anlz_fibwetdry	<i>Identify Fecal Indicator Bacteria samples as coming from a 'wet' or 'dry' time period</i>
----------------	--

---

**Description**

Identify Fecal Indicator Bacteria samples as coming from a 'wet' or 'dry' time period

**Usage**

```
anlz_fibwetdry(fibdata, precipdata, temporal_window = 2, wet_threshold = 0.5)
```

**Arguments**

fibdata	input data frame
precipdata	input data frame as returned by <a href="#">read_importrain</a> . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object <a href="#">catchprecip</a> has this data from 1995-2023 for select stations.
temporal_window	numeric, number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric, inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period

**Details**

This function allows the user to specify a threshold for declaring a sample to be taken after an important amount of rain over an important amount of days, and declaring it to be 'wet'. This is of interest because samples taken after significant precipitation (definitions of this vary, which is why the user can specify desired thresholds) are more likely to exceed relevant bacterial thresholds. Identifying samples as 'wet' or not allows for calculation of further indices for wet and dry subsets of samples.

**Value**

a data frame; the original fibdata data frame with three additional columns. `rain_sampleDay` is the total rain (inches) on the day of sampling, `rain_total` is the total rain (inches) for the period of time defined by `temporal_window`, and `wet_sample` is logical, indicating whether the rainfall for that station's catchment exceeded the amount over the time period specified in args.

**Examples**

```
entero_wetdry <- anz_fibwetdry(enterodata, catchprecip)
head(entero_wetdry)
```

---

anz\_hmpreport

*Evaluate Habitat Master Plan progress for report card*


---

**Description**

Evaluate Habitat Master Plan progress for report card

**Usage**

```
anz_hmpreport(acres, subtacres, hmptrgs)
```

**Arguments**

<code>acres</code>	data.frame for intertidal and supratidal land use and cover of habitat types for each year of data
<code>subtacres</code>	data.frame for subtidal cover of habitat types for each year of data
<code>hmptrgs</code>	data.frame of Habitat Master Plan targets and goals

**Details**

The relevant output columns are `targeteval` and `goaleval` that indicate numeric values as -1 (target not met, trending below), 0 (target met, trending below), 0.5 (target not met, trending above), and 1 (target met, trending above).

Columns in the output are as follows:

**year** Year of the assessment

**metric** Habitat type assessed

**Acres** Coverage estimate for the year

**lacres** Coverage estimate for the previous set of available data

**lyr** Year for the previous set of available data

**category** Strata for the habitat type

**Target** 2030 target for the habitat type from the Habitat Master Plan

**Goal** 2050 goal for the habitat type from the Habitat Master Plan

**acresdiff** Difference in acres for the current year and the previous set of available data

**yeardiff** Difference in years for the current year and the previous set of available data

**changerate** Acreage change per year for the current year relative to the previous set of available data

**targetrate** Annual rate required to achieve the 2030 target

**goalrate** Annual rate required to achieve the 2050 goal

**targetprop** Proportion of target met for the current year

**goalprop** Proportion of goal met for the current year

**targeteval** A number indicating target status of the current year for the report card

**goaleval** A number indicating goal status of the current year for the report card

The numbers in `targeteval` and `goaleval` are one of four values as -1, 0, 0.5, and 1. These numbers define the status for the assessment year:

**-1** target or goal not met, trending below

**0** target or goal met, trending below

**0.5** target or goal not met, trending above

**1** target or goal met, trending above

### Value

A summarized `data.frame` appropriate for creating a report card

### Examples

```
# view summarized data for report card
anlz_hmpreport(acres, subtacres, hmptrgs)
```

---

anlz_hydroload	<i>Estimate hydrological estimates and adjustment factors for bay segments</i>
----------------	--

---

### Description

Estimate hydrological estimates and adjustment factors for bay segments

### Usage

```
anlz_hydroload(yrs, noaa_key = NULL, trace = FALSE)
```

### Arguments

yrs	numeric vector indicating years to return
noaa_key	user-supplied NOAA key, see details
trace	logical indicating if function progress is printed in the consol

### Details

This function uses rainfall and streamflow data from NOAA and USGS and requires an API key. See the "Authentication" section under the help file for ncdc in the defunct rnoaa package. This key can be added to the R environment file and called for later use, see the examples.

These estimates are used in annual compliance assessment reports produced by the Tampa Bay Nitrogen Management Consortium. Load estimates and adjustment factors are based on regression models in <https://drive.google.com/file/d/11NT0NQ2WbPO6pVZaD7P7Z6qjcwO1jxHw/view?usp=drivesdk>

### Value

A data frame with hydrological load estimates by bay segments for the requested years

### Examples

```
## Not run:
# this function requires an API key
# save it to the R environment file (only once)
# save the key, do only once
cat("NOAA_KEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\n",
    file=file.path(normalizePath("~/"), ".Renviro"),
    append=TRUE)

# retrieve the key after saving, may need to restart R
noaa_key <- Sys.getenv('NOAA_key')

# get estimates for 2021
anlz_hydroload(2021, noaa_key)

## End(Not run)
```



---

anlz_iwrraw	<i>Format raw IWR data</i>
-------------	----------------------------

---

**Description**

Format raw IWR data

**Usage**

```
anlz_iwrraw(iwrraw, tidalcreeks, yr = 2023)
```

**Arguments**

iwrraw	FDEP impaired waters rule data base as <a href="#">data.frame</a>
tidalcreeks	<a href="#">sf</a> object for population of tidal creeks
yr	numeric for reference year to evaluate, scores are based on ten years of data using yr as the last year

**Details**

The function subsets the raw IWR data for ten years starting at the value in yr and subsets by the creek population in [tidalcreeks](#). Select water quality parameters in masterCode are filtered and some of the names are combined for continuity.

**Value**

A [data.frame](#) with the formatted data

**Examples**

```
anlz_iwrraw(iwrraw, tidalcreeks, yr = 2023)
```

---

anlz_refs	<i>Convert references csv to bib</i>
-----------	--------------------------------------

---

**Description**

Convert references csv to bib

**Usage**

```
anlz_refs(path)
```

**Arguments**

path	chr string of path to reference csv file or data frame object
------	---

**Value**

A data frame with references formatted as bib entries

**Examples**

```
# input and format
path <- 'https://raw.githubusercontent.com/tbep-tech/tbep-refs/master/tbep-refs.csv'
bibs <- anlz_refs(path)

## Not run:
# save output
writeLines(bibs, 'formatted.bib')

## End(Not run)
```

---

anlz\_sedimentaddtot    *Add contaminant totals to sediment data*

---

**Description**

Add contaminant totals to sediment data

**Usage**

```
anlz_sedimentaddtot(
  sedimentdata,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  param = NULL,
  pelave = TRUE
)
```

**Arguments**

sedimentdata	input sediment data.frame as returned by <a href="#">read_importsediment</a>
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
param	optional character string of a parameter to filter the results
pelave	logical indicating if output is used for <a href="#">anlz_sedimentpel</a>

**Details**

This function adds totals to the `sedimentdata` input for total PCBs, total DDT, total LMW PAH, total HMW PAH, and total PAH. Appropriate TEL/PEL values for the totals are also added.

**Value**

A `data.frame` object similar to the input, but filtered by the arguments and contaminant totals added. Replicate samples are also removed.

**Examples**

```
anlz_sedimentaddtot(sedimentdata)
```

---

<code>anlz_sedimentave</code>	<i>Get average concentrations for a sediment parameter by bay segment</i>
-------------------------------	---

---

**Description**

Get average concentrations for a sediment parameter by bay segment

**Usage**

```
anlz_sedimentave(
  sedimentdata,
  param,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP"
)
```

**Arguments**

<code>sedimentdata</code>	input sediment <code>data.frame</code> as returned by <a href="#">read_importsediment</a>
<code>param</code>	chr string for which parameter to plot
<code>yrrng</code>	numeric vector indicating min, max years to include, use single year for one year of data
<code>bay_segment</code>	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
<code>funding_proj</code>	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"

**Details**

Summaries for all bay segments are returned by default. The averages and confidence intervals are specific to the year ranges in `yrrng`.

**Value**

A data.frame object with average sediment concentrations for the selected parameter by bay segment

**See Also**

[show\\_sedimentave](#)

**Examples**

```
anz_sedimentave(sedimentdata, param = 'Arsenic')
```

---

anz_sedimentpel	<i>Get sediment PEL ratios at stations in Tampa Bay</i>
-----------------	---

---

**Description**

Get sediment PEL ratios at stations in Tampa Bay

**Usage**

```
anz_sedimentpel(
  sedimentdata,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP"
)
```

**Arguments**

sedimentdata	input sediment data.frame as returned by <a href="#">read_importsediment</a>
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"

**Details**

Average PEL ratios for all contaminants graded from A to F for benthic stations monitored in Tampa Bay are estimated. The PEL is a measure of how likely a contaminant is to have a toxic effect on invertebrates that inhabit the sediment. The PEL ratio is the contaminant concentration divided by the Potential Effects Levels (PEL) that applies to a contaminant, if available. Higher ratios and lower grades indicate sediment conditions that are likely unfavorable for invertebrates. The station average combines the PEL ratios across all contaminants measured at a station and the grade applies to the average.

The grade breaks for the PEL ratio are 0.00756, 0.02052, 0.08567, and 0.28026, with lower grades assigned to the higher breaks.

### Value

A `data.frame` object with average PEL ratios and grades at each station

### See Also

[show\\_sedimentpelmap](#)

### Examples

```
anlz_sedimentpel(sedimentdata)
```

---

`anlz_sedimentpelave` *Get average concentrations for a sediment parameter by bay segment*

---

### Description

Get average concentrations for a sediment parameter by bay segment

### Usage

```
anlz_sedimentpelave(
  sedimentdata,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP"
)
```

### Arguments

<code>sedimentdata</code>	input sediment <code>data.frame</code> as returned by <a href="#">read_importsediment</a>
<code>yrrng</code>	numeric vector indicating min, max years to include, use single year for one year of data
<code>bay_segment</code>	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
<code>funding_proj</code>	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"

### Details

Summaries for all bay segments are returned by default. The averages and confidence intervals are specific to the year ranges in `yrrng`.

**Value**

A data.frame of the average of the average PEL ratios for each bay segment

**See Also**

[show\\_sedimentpelave](#)

**Examples**

```
anlz_sedimentpelave(sedimentdata)
```

---

anlz_tbbimed	<i>Get annual medians of Tampa Bay Benthic Index scores by bay segment</i>
--------------	--

---

**Description**

Get annual medians of Tampa Bay Benthic Index scores by bay segment

**Usage**

```
anlz_tbbimed(
  tbbisocr,
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"),
  rev = FALSE,
  yrrng = c(1993, 2022)
)
```

**Arguments**

tbbisocr	input data frame as returned by <a href="#">anlz_tbbisocr</a>
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"
rev	logical if factor levels for bay segments are reversed
yrrng	numeric indicating year ranges to evaluate

**Details**

Additional summaries are provided for the entire bay, as a summary across categories ("All") and a summary weighted across the relative sizes of each bay segment ("All (wt)").

Only sampling funded by TBEP and as part of the routine EPC benthic monitoring program are included in the final categories.

**Value**

A data frame of annual medians by bay segment

**Examples**

```
tbbiscr <- anlz_tbbiscr(benthicdata)
anzl_tbbimed(tbbiscr)
```

---

anlz_tbbiscr	<i>Get Tampa Bay Benthic Index scores</i>
--------------	---

---

**Description**

Get Tampa Bay Benthic Index scores

**Usage**

```
anlz_tbbiscr(benthicdata)
```

**Arguments**

benthicdata    nested [tibble](#) formatted from [read\\_importbenthic](#)

**Details**

This function calculates scores for the TBBi based on station, taxa, and field sample data. The total TBBi scores are returned as TBBi and TBBiCat, where the latter is a categorical description of the scores.

**Value**

A single data frame of TBBi scores for each site.

**Examples**

```
anlz_tbbiscr(benthicdata)
```

---

anlz_tbniaive	<i>Get annual averages of Tampa Bay Nekton Index scores by bay segment</i>
---------------	--

---

**Description**

Get annual averages of Tampa Bay Nekton Index scores by bay segment

**Usage**

```
anlz_tbniave(
  tbniscr,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  rev = FALSE,
  perc = c(32, 46)
)
```

**Arguments**

tbniscr	input data frame as returned by <a href="#">anlz_tbniscr</a>
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB"
rev	logical if factor levels for bay segments are reversed
perc	numeric values indicating break points for score categories

**Value**

A data frame of annual averages by bay segment

**Examples**

```
tbniscr <- anlz_tbniscr(fimdata)
anlz_tbniave(tbniscr)
```

---

anlz\_tbnimet

*Get all raw metrics for Tampa Bay Nekton Index*

---

**Description**

Get all raw metrics for Tampa Bay Nekton Index

**Usage**

```
anlz_tbnimet(fimdata, all = FALSE)
```

**Arguments**

fimdata	data.frame formatted from <code>read_importfim</code>
all	logical indicating if only TBNI metrics are returned (default), otherwise all are calculated

**Details**

All raw metrics are returned in addition to those required for the TBNI. Each row shows metric values for a station, year, and month where fish catch data were available.



**Value**

A data frame of raw metrics in wide format. If `all = TRUE`, all metrics are returned, otherwise only `NumTaxa`, `BenthicTaxa`, `TaxaSelect`, `NumGuilds`, and `Shannon` are returned.

**Examples**

```
anlz_tbnimet(fimdata)
```

---

<code>anlz_tbniscr</code>	<i>Get Tampa Bay Nekton Index scores</i>
---------------------------	--

---

**Description**

Get Tampa Bay Nekton Index scores

**Usage**

```
anlz_tbniscr(fimdata, raw = TRUE)
```

**Arguments**

<code>fimdata</code>	data.frame formatted from <a href="#">read_importfim</a>
<code>raw</code>	logical indicating if raw metric values are also returned

**Details**

This function calculates raw and scored metrics for the TBNI, including `NumTaxa`, `BenthicTaxa`, `TaxaSelect`, `NumGuilds`, and `Shannon`. The total TBNI score is returned as `TBNI_Score`.

**Value**

A data frame of metrics and TBNI scores in wide format.

**See Also**

[anlz\\_tbnimet](#)

**Examples**

```
anlz_tbniscr(fimdata)
```

---

anlz\_tdlcrk      *Estimate tidal creek report card scores*

---

### Description

Estimate tidal creek report card scores

### Usage

```
anlz_tdlcrk(tidalcreeks, iwrraw, tidtrgs = NULL, yr = 2023)
```

### Arguments

tidalcreeks	<a href="#">sf</a> object for population of tidal creeks
iwrraw	FDEP impaired waters data base as <a href="#">data.frame</a>
tidtrgs	optional <a href="#">data.frame</a> for tidal creek nitrogen targets, defaults to <a href="#">tidaltargets</a>
yr	numeric for reference year to evaluate, scores are based on the planning period beginning ten years prior to this date

### Value

A [data.frame](#) with the report card scores for each creek, as prioritize, investigate, caution, monitor, or no data

### Examples

```
anlz_tdlcrk(tidalcreeks, iwrraw, yr = 2023)
```

---

anlz\_tdlcrkindic      *Analyze tidal creek water quality indicators*

---

### Description

Estimate tidal creek water quality indicators to support report card scores

### Usage

```
anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2023, radar = FALSE)
```

### Arguments

tidalcreeks	<a href="#">sf</a> object for population of tidal creeks
iwrraw	FDEP impaired waters rule data base as <a href="#">data.frame</a>
yr	numeric for reference year to evaluate, scores are based on the planning period beginning ten years prior to this date
radar	logical indicating if output is for <a href="#">show_tdlcrkradar</a> , see details

## Details

Annual geometric means for additional water quality data available at each wbid, JEI combination. Florida trophic state index values are also estimated where data are available.

Nitrogen ratios are estimated for JEIs that cover source (upstream, freshwater) and tidal (downstream) WBIDs, defined as the ratio of concentrations between the two (i.e., ratios > 1 mean source has higher concentrations). Nitrogen ratios for a given year reflect the ratio of the median nitrogen concentrations when they were measured in both a source and tidal segment during the same day. Note that a ratio of one can be obtained if both the source and tidal segments are at minimum detection.

Indicators for years where more than 10% of observations exceed DO saturation criteria are also estimated. The `do_bnm1` and `do_prop` columns show a 1 or 0 for a given year to indicate if more than ten percent of observations were below DO percent saturation of 42. The first column is based on a binomial probability exceedance that takes into account the number of observations in the year and the second column is based on a simple proportional estimate from the raw data.

If `radar = TRUE`, output is returned in a format for use with `show_tdlcrkradar`. Specifically, results are calculated as the percentage of years where an indicator exceeds a relevant threshold. This only applies to the marine WBIDs of the tidal creeks (Florida DEP class 2, 3M). Six indicators are returned with percentage exceedances based on total nitrogen (`tn_ind`) greater than 1.1 mg/L, chlorophyll (`chl_a_ind`) greater than 11 ug/L, trophic state index (`tsi_ind`) greater than 55 (out of 100), nitrate/nitrite ratio between marine and upstream segments (`nox_ind`) greater than one, chlorophyll and total nitrogen ratios > 15, and percentage of years more where than ten percent of observations were below DO percent saturation of 42.

## Value

A `data.frame` with the indicator values for each tidal creek

## Examples

```
dat <- anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2023)
head(dat)
```

---

anlz_transectave	<i>Get annual averages of seagrass frequency occurrence by bay segments and year</i>
------------------	--

---

## Description

Get annual averages of seagrass frequency occurrence by bay segments and year

## Usage

```
anlz_transectave(
  transectocc,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCB"),
  total = TRUE,
```

```

  yrrng = c(1998, 2023),
  rev = FALSE
)
```

### Arguments

transectocc	data frame returned by <a href="#">anlz_transectocc</a>
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB", "BCB"
total	logical indicating if average frequency occurrence is calculated for the entire bay across segments
yrrng	numeric indicating year ranges to evaluate
rev	logical if factor levels for bay segments are reversed

### Details

The `focat` column returned in the results shows a color category based on arbitrary breaks of the frequency occurrence estimates (`foest`) at 25, 50, and 75 percent. These don't necessarily translate to any ecological breakpoints.

### Value

A data frame of annual averages by bay segment

### Examples

```

## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
anlz_transectave(transectocc)
```

---

anlz_transectavespp	<i>Get annual averages of seagrass frequency occurrence by bay segments, year, and species</i>
---------------------	--

---

### Description

Get annual averages of seagrass frequency occurrence by bay segments, year, and species

**Usage**

```

anlz_transectavespp(
  transectocc,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCB"),
  yrrng = c(1998, 2023),
  species = c("Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa",
    "Dapis"),
  total = TRUE,
  by_seg = FALSE
)

```

**Arguments**

transectocc	data frame returned by <a href="#">anlz_transectocc</a>
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB", "BCB"
yrrng	numeric indicating year ranges to evaluate
species	chr string of species to summarize, one to many of "Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa", "Dapis"
total	logical indicating if total frequency occurrence for all species is also returned
by_seg	logical indicating if separate results by bay segments are retained

**Details**

Frequency occurrence estimates are averaged across segments in bay\_segment if by\_seg = F, i.e., separate results by location are not returned. Results are retained by bay segment if by\_seg = T. Also note that totals across species (total = T) are not returned if by\_seg = T.

**Value**

A data frame of annual averages by bay segment

**Examples**

```

## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
anlz_transectavespp(transectocc)

```

---

anlz_transectocc	<i>Get seagrass average abundance and occurrence across transects</i>
------------------	---

---

### Description

Get seagrass average abundance and occurrence across transects

### Usage

```
anlz_transectocc(transect)
```

### Arguments

transect            data frame returned by [read\\_transect](#)

### Details

Abundance and frequency occurrence are estimated as in Sherwood et al. 2017, equations 1 and 2. In short, frequency occurrence is estimated as the number of instances a species was observed along a transect divided by the number of placements along a transect and average abundance was estimated as the sum of species-specific Braun-Blanquet scores divided by the number of placements along a transect. The estimates are obtained for all seagrass species including *Caulerpa* (attached macroalgae) and *Dapis* (cyanobacteria), whereas all attached and drift algae species are aggregated. Drift or attached macroalgae and cyanobacteria (*Dapis*) estimates may not be accurate prior to 2021.

### Value

A data frame with abundance and frequency occurrence estimates aggregated by species, transect, and date. The nsites column is the total number of placements that were sampled along a transect for a particular date.

### References

Sherwood, E.T., Greening, H.S., Johansson, J.O.R., Kaufman, K., Raulerson, G.E. 2017. Tampa Bay (Florida, USA): Documenting seagrass recovery since the 1980's and reviewing the benefits. *Southeastern Geographer*. 57(3):294-319.

### Examples

```
## Not run:  
transect <- read_transect()  
  
## End(Not run)  
anlz_transectocc(transect)
```

---

anlz_yrattain	<i>Get attainment categories for a chosen year</i>
---------------	--

---

**Description**

Get attainment categories for a chosen year

**Usage**

```
anlz_yrattain(epcdata, yrssel, partialyr = FALSE)
```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importtwq</a>
yrssel	numeric indicating chosen year
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

**Value**

A data.frame for the chosen year and all bay segments showing the bay segment averages for chlorophyll concentration, light attenuations, segment targets, and attainment categories.

**Examples**

```
# defaults to current year
anlz_yrattain(epcdata, yrssel = 2023)
```

---

benthicdata	<i>Benthic data for the Tampa Bay Benthic Index current as of 03132024</i>
-------------	--

---

**Description**

Benthic data for the Tampa Bay Benthic Index current as of 03132024

**Usage**

```
benthicdata
```

**Format**

A nested [tibble](#) with 3 rows and 2 variables:

**name** chr identifying the dataset as stations, fieldsamples, or taxacounts

**value** list of dataframes for each dataset

**Details**

Index the corresponding list element in the value column to view each dataset. For example, the stations data in the first row can be viewed as `benthicdata$value[[1]]`.

**Examples**

```
## Not run:
# location to download data
path <- '~/Desktop/benthic.zip'

# load and assign to object
benthicdata <- read_importbenthic(path, download_latest = TRUE, remove = TRUE)

save(benthicdata, file = 'data/benthicdata.RData', compress = 'xz')

## End(Not run)
```

---

bsmap	<i>Terrain basemap</i>
-------	------------------------

---

**Description**

Terrain basemap

**Usage**

```
bsmap
```

**Format**

A `ggmap` object

---

catchpixels	<i>Catchments and radar pixels (for precip) of selected Enterococcus stations</i>
-------------	---

---

**Description**

Catchments and radar pixels (for precip) of selected Enterococcus stations

**Usage**

```
catchpixels
```



**Format**

A data frame with two columns:

**station** a character string of the Water Quality Portal station name

**pixel** a number; each row is a single pixel from the Southwest Florida Water Management District's radar-estimated rainfall shapefile

**Details**

Generated by data-raw/catchpixels-raw.R (view on github: <https://github.com/tbep-tech/tbepools/blob/master/data-raw/catchpixels-raw.R>)

---

catchprecip

*Daily precip by catchment for selected Enterococcus stations*

---

**Description**

Daily precip by catchment for selected Enterococcus stations

**Usage**

catchprecip

**Format**

A data frame with three columns:

**station** a character string of the Water Quality Portal station name

**date** a date

**rain** a number; inches of rain for that date, averaged across all pixels in the station's catchment

**Details**

Daily precipitation data from 1995-2023, provided by the Southwest Florida Water Management District, were downloaded at the pixel level and averaged to the catchment level for key Enterococcus sampling stations. Generated by data-raw/catchprecip-raw.R (view on github: <https://github.com/tbep-tech/tbepools/blob/master/data-raw/catchprecip-raw.R>)

**Source**

Southwest Florida Management District radar-estimated daily rainfall data, [ftp://ftp.swfwmd.state.fl.us/pub/radar\\_rainfall/Daily\\_Data/](ftp://ftp.swfwmd.state.fl.us/pub/radar_rainfall/Daily_Data/)

---

enterodata

*Enterococcus data from 53 key Enterococcus stations since 1995*

---

### Description

Enterococcus data from 53 key Enterococcus stations since 1995

### Usage

enterodata

### Format

A data frame with 6266 rows and 16 columns:

date date, sample date

yr numeric, year of sample date

mo numeric, month of sample date

time character, sample time

time\_zone character, sample time zone

long\_name character, long name of bay segment subwatershed

bay\_segment character, short name of bay segment subwatershed

station character, sample station

entero numeric, Enterococcus concentration

entero\_censored logical, whether entero value was below the laboratory MDL, minimum detection limit

MDL numeric, minimum detection limit at the time of processing

entero\_units character, units of measurement for entero

qualifier qualifier codes associated with sample

LabComments lab comments on sample

Latitude numeric, latitude in decimal degrees

Longitude numeric, longitude in decimal degrees

### Details

A sample dataset containing Enterococcus from 53 stations in the TBEP watershed from 1995-2023. Generated by data-raw/enterodata-raw.R (view on github: <https://github.com/tbep-tech/tbep-tools/blob/master/data-raw/enterodata-raw.R>)

### Source

Water Quality Portal, <https://waterqualitydata.us>

---

epcdata	<i>All bay data as of 20240201</i>
---------	------------------------------------

---

**Description**

All bay data as of 20240201

**Usage**

epcdata

**Format**

A data frame with 27961 rows and 26 variables:

**bay\_segment** chr  
**epchc\_station** num  
**SampleTime** POSIXct  
**yr** num  
**mo** num  
**Latitude** num  
**Longitude** num  
**Total\_Depth\_m** num  
**Sample\_Depth\_m** num  
**tn** num  
**tn\_q** chr  
**sd\_m** num  
**sd\_raw\_m** num  
**sd\_q** chr  
**chla** num  
**chla\_q** chr  
**Sal\_Top\_ppth** num  
**Sal\_Mid\_ppth** num  
**Sal\_Bottom\_ppth** num  
**Temp\_Water\_Top\_degC** num  
**Temp\_Water\_Mid\_degC** num  
**Temp\_Water\_Bottom\_degC** num  
**Turbidity\_JTU-NTU** num  
**Turbidity\_Q** num  
**Color\_345\_F45\_PCU** num  
**Color\_345\_F45\_Q** num

## Examples

```
## Not run:
xlsx <- '~/Desktop/epcdata.xlsx'
epcdata <- read_importwq(xlsx, download_latest = TRUE)

nrow(epcdata)
ncol(epcdata)

save(epcdata, file = 'data/epcdata.RData', compress = 'xz')

## End(Not run)
```

---

fibdata

*All Fecal Indicator Bacteria (FIB) data as of 20240814*

---

## Description

All Fecal Indicator Bacteria (FIB) data as of 20240814

## Usage

fibdata

## Format

A data frame with 77526 rows and 18 variables:

**area** chr  
**epchc\_station** num  
**class** chr  
**SampleTime** POSIXct  
**yr** num  
**mo** num  
**Latitude** num  
**Longitude** num  
**Total\_Depth\_m** num  
**Sample\_Depth\_m** num  
**ecoli** num  
**ecoli\_q** chr  
**entero** num  
**entero\_q** chr  
**fcolid** num  
**fcolid\_q** chr  
**totcol** num  
**totcol\_q** chr

**Examples**

```
## Not run:
xlsx <- tempfile(fileext = '.xlsx')
fibdata <- read_importfib(xlsx, download_latest = TRUE)

nrow(fibdata)
ncol(fibdata)

save(fibdata, file = 'data/fibdata.RData', compress = 'xz')

file.remove(xlsx)

## End(Not run)
```

fimdاتا

*FIM data for Tampa Bay Nekton Index current as of 07092024***Description**

FIM data for Tampa Bay Nekton Index current as of 07092024

**Usage**

```
fimdاتا
```

**Format**

A data frame with 52042 rows and 19 variables:

**Reference** chr  
**Sampling\_Date** Date  
**Latitude** num  
**Longitude** num  
**Zone** chr  
**Grid** int  
**NODCCODE** chr  
**Year** num  
**Month** num  
**Total\_N** num  
**ScientificName** chr  
**Include\_TB\_Index** chr  
**Hab\_Cat** chr  
**Est\_Cat** chr  
**Est\_Use** chr

**Feeding\_Cat** chr  
**Feeding\_Guild** chr  
**Selected\_Taxa** chr  
**bay\_segment** chr

### Examples

```
## Not run:  
csv <- '~/Desktop/TampaBay_NektonIndexData.csv'  
  
fimdata <- read_importfim(csv, download_latest = TRUE)  
  
save(fimdata, file = 'data/fimdata.RData', compress = 'xz')  
  
## End(Not run)
```

---

fimstations

*Spatial data object of FIM stations including Tampa Bay segments*

---

### Description

Spatial data object of FIM stations including Tampa Bay segments

### Usage

```
fimstations
```

### Format

A simple features [sf](#) object (POINT) with 7771 features and 2 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs

**Reference** num  
**bay\_segment** chr

### Examples

```
## Not run:  
# file path  
csv <- '~/Desktop/fimdata.csv'  
  
# load and assign to object  
fimstations <- read_importfim(csv, download_latest = FALSE, locs = TRUE)  
save(fimstations, file = 'data/fimstations.RData', compress = 'xz')  
  
## End(Not run)
```

---

hmptrgs	<i>Habitat Master Plan targets and goals</i>
---------	--

---

**Description**

Habitat Master Plan targets and goals

**Usage**

```
hmptrgs
```

**Format**

A data frame showing 2030 targets and 2050 goals

**Examples**

```
## Not run:
library(dplyr)

load(url("https://github.com/tbep-tech/hmpu-workflow/raw/master/data/trgs.RData"))

hmptrgs <- trgs %>%
  rename(Goal2050 = Target2050)

save(hmptrgs, file = 'data/hmptrgs.RData', compress = 'xz')

## End(Not run)
```

---

iwrraw	<i>FDEP IWR run 65</i>
--------	------------------------

---

**Description**

Florida Department of Environmental Protection, Impaired Waters Rule, Run 65

**Usage**

```
iwrraw
```

**Format**

A data frame 537946 rows and 11 variables

**Details**

File was created using workflow at [https://tbep-tech.github.io/tidalcreek-stats/Creek\\_select\\_tbeptools](https://tbep-tech.github.io/tidalcreek-stats/Creek_select_tbeptools), example below is old and for Run 61.

## Examples

```
## Not run:
library(dplyr)

load(file = '../02_DOCUMENTS/tidal_creeks/iwrraw_run61.RData')
iwrraw <- sf::st_set_geometry(iwrraw, NULL) %>%
  rename(JEI = jei)
save(iwrraw, file = 'data/iwrraw.RData', compress = 'xz')

## End(Not run)
```

---

phytodata

*Phytoplankton data current as of 03202024*

---

## Description

Phytoplankton data current as of 03202024

## Usage

```
phytodata
```

## Format

A nested [tibble](#) with 21143 rows and 8 variables:

```
epchc_station chr
Date Date
name chr
units chr
count num
yrqrt Date
yr num
mo Ord.factor
```

## Examples

```
## Not run:
# location to download data
path <- tempfile(fileext = '.xlsx')

# load and assign to object
phytodata <- read_importphyto(path, download_latest = TRUE)

nrow(phytodata)
ncol(phytodata)
```



```

save(phytodata, file = 'data/phytodata.RData', compress = 'xz')

file.remove(path)

## End(Not run)

```

---

read_dlcurrent	<i>Download latest file from epchc.org</i>
----------------	--

---

### Description

Download latest file from epchc.org

### Usage

```
read_dlcurrent(locin, download_latest = TRUE, urlin)
```

### Arguments

locin	chr string path for local file, to overwrite it not current
download_latest	logical to download latest file regardless of local copy
urlin	url for file location

### Details

The local copy is checked against a temporary file downloaded from the location specified by urlin. The local file is replaced with the downloaded file if the MD5 hashes are different.

### Value

The local copy specified in the path by locin is overwritten by the new file if the local copy is not current or download\_latest = TRUE. The function does nothing if download\_latest = FALSE.

### Examples

```

## Not run:
locin <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'
urlin1 <- 'https://epcbocc.sharepoint.com/:x:/s/Share/'
urlin2 <- 'EYXZ5t16U1FGk1rzIU91VogBa8U371h8z_Hftf2KJISSHg?e=8r1SUL'
urlin <- paste0(urlin1, urlin2, '&download=1')
read_dlcurrent(locin = locin, urlin = urlin)

## End(Not run)

```

---

read\_formbenthic      *Format benthic data for the Tampa Bay Benthic Index*

---

### Description

Format benthic data for the Tampa Bay Benthic Index

### Usage

```
read_formbenthic(pathin)
```

### Arguments

pathin              A path to unzipped csv files with base tables used to calculate benthic index

### Details

Function is used internally within [read\\_importbenthic](#)

### Value

A nested [tibble](#) of station, field sample, and taxa data

### Examples

```
## Not run:

# location to download data
path <- '~/Desktop/benthic.zip'

# download
urlin1 <- 'https://epcbocc.sharepoint.com/:x:/s/Share/'
urlin2 <- 'EQUCWBUwCNDGuMREYAyAD1gBKC98mYtCHMWX0FYLRbT4KA?e=nDfnnQ'
urlin <- paste0(urlin1, urlin2, '&download=1')
read_dlcurrent(path, download_latest = TRUE, urlin = urlin)

# unzip
tmppth <- tempfile()
utils::unzip(path, exdir = tmppth, overwrite = TRUE)

# format benthic data
read_formbenthic(pathin = tmppth)

# remove temporary path
unlink(tmppth, recursive = TRUE)

## End(Not run)
```

---

read_formfib	<i>Format Fecal Indicator Bacteria (FIB) data</i>
--------------	---

---

### Description

Format Fecal Indicator Bacteria (FIB) data

### Usage

```
read_formfib(datin, all = FALSE)
```

### Arguments

datin	input data.frame loaded from <a href="#">read_importepc</a>
all	logical indicating if all stations with FIB data are returned, default is FALSE

### Details

Formats input data from [read\\_importepc](#) appropriate for FIB results, see the details in [read\\_importfib](#) for more more information.

### Value

A lightly formatted data.frame with FIB data

### See Also

[read\\_importfib](#), [read\\_importepc](#)

### Examples

```
## Not run:  
# file path  
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'  
  
# load raw data and assign to object  
epcall <- read_importepc(xlsx, download_latest = T)  
  
# final formatting  
fibdata <- read_formfib(epcall)  
  
## End(Not run)
```

---

read_formfim	<i>Format FIM data for the Tampa Bay Nekton Index</i>
--------------	---

---

### Description

Format FIM data for the Tampa Bay Nekton Index

### Usage

```
read_formfim(datin, locs = FALSE)
```

### Arguments

datin	input data.frame loaded from <a href="#">read_importfim</a>
locs	logical indicating if a spatial features object is returned with locations of each FIM sampling station

### Details

Function is used internally within [read\\_importfim](#)

### Value

A formatted data.frame with FIM data if locs = FALSE, otherwise a simple features object if locs = TRUE

### See Also

[read\\_importfim](#)

### Examples

```
# file path
csv <- url('https://raw.githubusercontent.com/tbep-tech/tbni-proc/master/data/TampaBay_NektonIndexData.csv')

datin <- read.csv(csv)

# load and assign to object
fimdata <- read_formfim(datin)
```

---

read_formphyto	<i>Format phytoplankton data</i>
----------------	----------------------------------

---

## Description

Format phytoplankton data

## Usage

```
read_formphyto(datin)
```

## Arguments

datin            input data.frame loaded from [read\\_importphyto](#)

## Details

Only seven taxonomic groups are summarized. Pyrodinium bahamense, Karenia brevis, Tripos hircus, Pseudo-nitzschia sp., and Pseudo-nitzschia pungens are retained at the species level. Bacillariophyta and Cyanobacteria are retained at the phylum level. All other taxa are grouped into an "other" category.

## Value

A formatted data.frame with phytoplankton count data

## See Also

[read\\_importphyto](#)

## Examples

```
## Not run:  
# file path  
xlsx <- '~/Desktop/phyto_data.xlsx'  
  
# load and assign to object  
phytodata <- read_importphyto(xlsx)  
  
## End(Not run)
```

---

read\_formsediment      *Format sediment data*

---

### Description

Format sediment data

### Usage

```
read_formsediment(pathin)
```

### Arguments

pathin                  A path to unzipped csv files with sediment result tables

### Details

Function is used internally within [read\\_importsediment](#)

### Value

A data.frame of sediment data for Tampa Bay

### Examples

```
## Not run:

# location to download data
path <- '~/Desktop/sediment.zip'

# download
urlin1 <- 'https://epcbocc.sharepoint.com/:x:/s/Share/'
urlin2 <- 'Ef9utuKCHD9LliarsOPKCJwB5kxgCObf0tY5x5wX20JQUA?e=DuTseb'
urlin <- paste0(urlin1, urlin2, '&download=1')
read_dlcurrent(path, download_latest = TRUE, urlin = urlin)

# unzip
tmppth <- tempfile()
utils::unzip(path, exdir = tmppth, overwrite = TRUE)

# format sediment data
read_formsediment(pathin = tmppth)

# remove temporary path
unlink(tmppth, recursive = TRUE)

## End(Not run)
```

---

read\_formtransect      *Format seagrass transect data from Water Atlas*

---

### Description

Format seagrass transect data from Water Atlas

### Usage

```
read_formtransect(jsn, training = FALSE, raw = FALSE)
```

### Arguments

jsn	A data frame returned from <a href="#">fromJSON</a>
training	logical if input are transect training data or complete database
raw	logical indicating if raw, unformatted data are returned, see details

### Details

Shoot density is reported as number of shoots per square meter and is corrected for the quadrat size entered in the raw data. Shoot density and blade height (cm) are based on averages across random observations at each transect point that are entered separately in the data form. Abundance is reported as a numeric value from 0 - 5 for Braun-Blanquet coverage estimates.

If raw = TRUE, the unformatted data are returned. The default is to use formatting that allows the raw data to be used with the downstream functions. The raw data may have extra information that may be of use outside of the plotting functions in this package.

### Value

data frame in long format

### Examples

```
library(jsonlite)

## Not run:
# all transect data
url <- 'http://dev.seagrass.wateratlas.usf.edu/api/assessments/all__use-with-care'
jsn <- fromJSON(url)
trndat <- read_formtransect(jsn)

## End(Not run)

# training transect data
url <- 'http://dev.seagrass.wateratlas.usf.edu/api/assessments/training'
jsn <- fromJSON(url)
trndat <- read_formtransect(jsn, training = TRUE)
```

---

read_formwq	<i>Format water quality data</i>
-------------	----------------------------------

---

### Description

Format water quality data

### Usage

```
read_formwq(datin, all = FALSE)
```

### Arguments

datin	input data.frame loaded from <a href="#">read_importepc</a>
all	logical indicating if all water quality parameters are returned, see details

### Details

Secchi data VOB depths or secchis < 0.5 ft from bottom are assigned NA, function is used internally within [read\\_importwq](#)

### Value

A lightly formatted data.frame with chloropyll and secchi observations

### See Also

[read\\_importwq](#), [read\\_importepc](#)

### Examples

```
## Not run:  
# file path  
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'  
  
# load raw data and assign to object  
epcall <- read_importepc(xlsx, download_latest = T)  
  
# final formatting  
epcdata <- read_formwq(epcall)  
  
## End(Not run)
```



---

read_formwqp	<i>Format data and station metadata from the Water Quality Portal</i>
--------------	---

---

### Description

Format data and station metadata from the Water Quality Portal

### Usage

```
read_formwqp(res, sta, org, type, trace = F)
```

### Arguments

res	A data frame containing results obtained from the API
sta	A data frame containing station metadata obtained from the API
org	chr string indicating the organization identifier
type	chr string indicating data type to download, one of "wq" or "fib"
trace	Logical indicating whether to display progress messages, default is FALSE

### Details

This function is used by [read\\_importwqp](#) to combine, format, and process data (res) and station metadata (sta) obtained from the Water Quality Portal for the selected county and data type. The resulting data frame includes the date, time, station identifier, latitude, longitude, variable name, value, unit, and quality flag.

### Value

A data frame containing formatted water quality and station metadata

### See Also

[read\\_importwqp](#)

### Examples

```
## Not run:
url <- list(
  Result = "https://www.waterqualitydata.us/data/Result/search?mimeType=csv&zip=no",
  Station = "https://www.waterqualitydata.us/data/Station/search?mimeType=csv&zip=no"
)

headers <- c(
  "Content-Type" = "application/json",
  "Accept" = "application/zip"
)
```

```

body <- list(
  organization = c("21FLMANA_WQX"),
  sampleMedia = c("Water"),
  characteristicType = c("Information", "Nutrient", "Biological, Algae, Phytoplankton,
    Photosynthetic Pigments"),
  providers = c("STORET"),
  siteType = c("Estuary")
)

res <- url[['Result']] %>%
  httr::POST(httr::add_headers(headers), body = jsonlite::toJSON(body)) %>%
  httr::content('text') %>%
  read.csv(text = .)

sta <- url[['Station']] %>%
  httr::POST(httr::add_headers(headers), body = jsonlite::toJSON(body)) %>%
  httr::content('text') %>%
  read.csv(text = .)

# combine and format
read_formwqp(res, sta, '21FLMANA_WQX', type = 'wq')

## End(Not run)

```

---

read\_importbenthic      *Download and import benthic data for Tampa Bay*

---

## Description

Download and import benthic data for Tampa Bay

## Usage

```
read_importbenthic(path, download_latest = FALSE, remove = FALSE)
```

## Arguments

path	chr string for local path where the zipped folder will be downloaded, must include .zip extension
download_latest	logical to download latest if a more recent dataset is available
remove	logical if the downloaded folder is removed after unzipping

## Details

This function downloads and unzips a folder of base tables used to calculate the benthic index from <https://epcbocc.sharepoint.com/:u:/s/Share/EQUQCWBuwCNDGuMREYAyAD1gBKC98mYtCHMWX0FYLRbT4KA?e=nDfnnQ&download=1> (viewable at <https://epcbocc.sharepoint.com/:f:/s/Share/EtOJfziTTa9F1iL1oR0b90sBRZU0NRC162hHjQ?e=4gUXgJ>).

Index the corresponding list element in the value column to view each dataset. For example, the stations data in the first row can be viewed as `benthicdata$value[[1]]`.

### Value

A nested tibble of station, taxa, and field sample data.

### Examples

```
## Not run:
# location to download data
path <- '~/Desktop/benthic.zip'

# load and assign to object
benthicdata <- read_importbenthic(path, download_latest = TRUE)

## End(Not run)
```

---

read_importentero	<i>Download Enterococcus data from the Water Quality Portal</i>
-------------------	---

---

### Description

Download Enterococcus data from the Water Quality Portal

### Usage

```
read_importentero(stas = NULL, startDate, endDate)
```

### Arguments

stas	character, a vector of stations. If NULL, defaults to all stations in <a href="#">catchprecip</a> .
startDate	character, starting date of observations as YYYY-MM-DD
endDate	character, ending date of observations as YYYY-MM-DD

### Details

Retrieves Enterococcus sample data from selected stations and date range from the Water Quality Portal, <https://www.waterqualitydata.us>

### Value

a data frame containing one row for each sample. Columns returned are:

date	date, sample date
yr	numeric, year of sample date
mo	numeric, month of sample date

time character, sample time  
 time\_zone character, sample time zone  
 long\_name character, long name of bay segment subwatershed  
 bay\_segment character, short name of bay segment subwatershed  
 station character, sample station  
 entero numeric, Enterococcus concentration  
 entero\_censored logical, whether entero value was below the laboratory MDL, minimum detection limit  
 MDL numeric, minimum detection limit at the time of processing  
 entero\_units character, units of measurement for entero  
 qualifier qualifier codes associated with sample  
 LabComments lab comments on sample  
 Latitude numeric, latitude in decimal degrees  
 Longitude numeric, longitude in decimal degrees

### Examples

```

## Not run:
# stations to download
stas <- c('21FLHILL_WQX-101',
          '21FLHILL_WQX-102',
          '21FLHILL_WQX-103')

# download and read the data
entero_in <- read_importentero(stas = stas, startDate = '2023-01-01', endDate = '2023-02-01')

head(entero_in)

## End(Not run)

```

---

read\_importepc

*Download and/or import local water quality file for internal use*

---

### Description

Download and/or import local water quality file for internal use

### Usage

```
read_importepc(xlsx, download_latest = FALSE, na = c("", "NULL"))
```

**Arguments**

xlsx	chr string path for local excel file, to overwrite if not current
download_latest	logical passed to <a href="#">read_dlcurrent</a> to download raw data and compare with existing in xlsx if available
na	chr vector of strings to interpret as NA, passed to <a href="#">read_xlsx</a>

**Details**

Loads the "RWMDDataSpreadsheet" worksheet from the file located at xlsx. The file is downloaded from [https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16U1FGk1rzIU91VogBa8U37lh8z\\_Hftf2KJISSHg?e=8r1SUL&download=1](https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16U1FGk1rzIU91VogBa8U37lh8z_Hftf2KJISSHg?e=8r1SUL&download=1). The files can be viewed at [https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv\\_6-t7kBUYaXiIqN0B1n2w57Z\\_V3kQ?e=NdZQcU](https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv_6-t7kBUYaXiIqN0B1n2w57Z_V3kQ?e=NdZQcU).

This function is used internally by [read\\_importwq](#) and [read\\_importfib](#) because both use the same source data from the Environmental Protection Commission of Hillsborough County.

**Value**

An unformatted data.frame from EPC

**See Also**

[read\\_importwq](#), [read\\_importfib](#)

**Examples**

```
## Not run:
# file path
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

# load and assign to object
epcall <- read_importepc(xlsx, download_latest = T)

## End(Not run)
```

---

read_importfib	<i>Load local water quality file for Fecal Indicator Bacteria (FIB)</i>
----------------	---

---

**Description**

Load local water quality file for Fecal Indicator Bacteria (FIB)

**Usage**

```
read_importfib(xlsx, download_latest = FALSE, na = c("", "NULL"), all = TRUE)
```

## Arguments

<code>xlsx</code>	chr string path for local excel file, to overwrite if not current
<code>download_latest</code>	logical passed to <code>read_dlcurrent</code> to download raw data and compare with existing in <code>xlsx</code> if available
<code>na</code>	chr vector of strings to interpret as NA, passed to <code>read_xlsx</code>
<code>all</code>	logical indicating if all stations with FIB data are returned, default is TRUE, see details

## Details

Loads the "RWMDDataSpreadsheet" worksheet from the file located at `xlsx`. The file is downloaded from [https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16U1FGk1rzIU91VogBa8U37lh8z\\_Hftf2KJISSHg?e=8r1SUL&download=1](https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16U1FGk1rzIU91VogBa8U37lh8z_Hftf2KJISSHg?e=8r1SUL&download=1). The files can be viewed at [https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv\\_6-t7kBUYaXiIqN0B1n2w57Z\\_V3kQ?e=NdZQcU](https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv_6-t7kBUYaXiIqN0B1n2w57Z_V3kQ?e=NdZQcU).

Returns FIB data including E. coli, Enterococcus, Fecal Coliform, and Total Coliform concentrations and waterbody class (freshwater as 1 or 3F, marine as 2 or 3M) for stations and sample dates, function is used internally within `read_importfib`

Values are returned for E. coli (`ecoli`), Enterococcus (`entero`), Fecal Coliform (`fcolif`), and Total Coliform (`totcol`). Values shown are # of colonies per 100 mL of water (`#/100mL`). Qualifier columns for each are also returned with the `_q` suffix. Qualifier codes can be interpreted from the source spreadsheet.

Concentrations noted with `<` or `>` in the raw data are reported as is, with only the numeric value shown. Samples with this notation can be determined from the qualifier columns.

If `all = FALSE`, only stations with `AreaName` in the source data as Hillsborough River, Hillsborough River Tributary, Alafia River, Alafia River Tributary, Lake Thonotosassa, Lake Thonotosassa Tributary, and Lake Roberta.

## Value

A data.frame of formatted FIB data.

## See Also

[read\\_formfib](#)

## Examples

```
## Not run:
# file path
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

# load and assign to object
fibdata <- read_importfib(xlsx, download_latest = T)

## End(Not run)
```

---

read_importfim	<i>Load local FIM data for the Tampa Bay Nekton Index</i>
----------------	---

---

### Description

Load local FIM data for the Tampa Bay Nekton Index

### Usage

```
read_importfim(csv, download_latest = FALSE, locs = FALSE)
```

### Arguments

csv	chr string path for local csv file, to overwrite if not current
download_latest	logical passed to <a href="#">read_dlcurrent</a> to download raw data and compare with existing in csv if available
locs	logical indicating if a spatial features object is returned with locations of each FIM sampling station

### Details

Data downloaded from [ftp://ftp.floridamarine.org/users/fim/tmac/NektonIndex/TampaBay\\_NektonIndexData.csv](ftp://ftp.floridamarine.org/users/fim/tmac/NektonIndex/TampaBay_NektonIndexData.csv).

### Value

A formatted data.frame with FIM data if locs = FALSE, otherwise a simple features object if locs = TRUE

### See Also

[read\\_formwq](#), [read\\_importphyto](#)

### Examples

```
## Not run:  
# file path  
csv <- '~/Desktop/fimraw.csv'  
  
# load and assign to object  
fimdata <- read_importfim(csv, download_latest = TRUE)  
  
## End(Not run)
```

---

read\_importphyto      *Load local phytoplankton cell count file*

---

### Description

Load local phytoplankton cell count file

### Usage

```
read_importphyto(xlsx, download_latest = FALSE, na = c("", "NULL"))
```

### Arguments

xlsx	chr string path for local excel file, to overwrite if not current
download_latest	logical passed to <a href="#">read_dlcurrent</a> to download raw data and compare with existing in xlsx if available
na	chr vector of strings to interpret as NA, passed to <a href="#">read_xlsx</a>

### Details

Phytoplankton cell count data downloaded from <https://epcbocc.sharepoint.com/:x:/s/Share/ETAfRQ5drmRHntDd1O8s3FQOIVDxrA?e=eSmtxD&download=1>

### Value

A data.frame of formatted water quality data.

### See Also

[read\\_importtwq](#)

### Examples

```
## Not run:  
# file path  
xlsx <- '~/Desktop/phyto_data.xlsx'  
  
# load and assign to object  
phytodata <- read_importphyto(xlsx, download_latest = T)  
  
## End(Not run)
```



---

read\_importrain      *Download daily precip data and summarise by station catchment*

---

### Description

Download daily precip data and summarise by station catchment

### Usage

```
read_importrain(curyr, catchpixels, mos = 1:12, quiet = T)
```

### Arguments

curyr	numeric for year
catchpixels	data.frame with columns named 'station' and 'pixel'. A data frame has been created for key Enterococcus stations, associating each station with all pixels in that station's catchment layer. This is the tbeptools object 'catchpixels'.
mos	numeric vector for months to download
quiet	logical for messages

### Details

Data from the Southwest Florida Water Management District's (SWFWMD) ftp site: [ftp://ftp.swfwmd.state.fl.us/pub/radar\\_ra](ftp://ftp.swfwmd.state.fl.us/pub/radar_ra)

### Value

data.frame with station, date, rain columns as a daily average (inches) for all pixels in a catchment

### Examples

```
## Not run:
read_importrain(2021, catchpixels, quiet = F)

## End(Not run)
```

---

read\_importsediment      *Download and import sediment data for Tampa Bay*

---

### Description

Download and import sediment data for Tampa Bay

### Usage

```
read_importsediment(path, download_latest = FALSE, remove = FALSE)
```

**Arguments**

path	chr string for local path where the zipped folder will be downloaded, must include .zip extension
download_latest	logical to download latest if a more recent dataset is available
remove	logical if the downloaded folder is removed after unzipping

**Details**

This function downloads and unzips a folder of results tables from <https://epcbocc.sharepoint.com/:u:/s/Share/Ef9utuKCHD9LliarsOPKCJwB5kxgC0bf0tY5x5wX20JQUA?e=DuTseb&download=1> (viewable at [https://epcbocc.sharepoint.com/:f:/s/Share/Et0JfziTTa9Flil1oR0b90sBRZU-n060fu\\_0NRC162hHjQ?e=4gUXgJ](https://epcbocc.sharepoint.com/:f:/s/Share/Et0JfziTTa9Flil1oR0b90sBRZU-n060fu_0NRC162hHjQ?e=4gUXgJ)).

The row entries for columns "BetweenTELPEL" and "ExceedsPEL" for rows where the "Qualifier" column is "U" or "T" (below detection, not detected) are assigned NA, regardless of the entry in the source data.

**Value**

A data.frame of sediment data for Tampa Bay

**Examples**

```
## Not run:
# location to download data
path <- '~/Desktop/sediment.zip'

# load and assign to object
sedimentdata <- read_importsediment(path, download_latest = TRUE)

## End(Not run)
```

---

read_importwq	<i>Load local water quality file</i>
---------------	--------------------------------------

---

**Description**

Load local water quality file

**Usage**

```
read_importwq(xlsx, download_latest = FALSE, na = c("", "NULL"), all = FALSE)
```

**Arguments**

xlsx	chr string path for local excel file, to overwrite if not current
download_latest	logical passed to <a href="#">read_dlcurrent</a> to download raw data and compare with existing in xlsx if available
na	chr vector of strings to interpret as NA, passed to <a href="#">read_xlsx</a>
all	logical indicating if all water quality parameters are returned, see details

**Details**

Loads the "RWMDDataSpreadsheet" worksheet from the file located at xlsx. The file is downloaded from [https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16UJFGk1rzIU91VogBa8U37lh8z\\_Hftf2KJISSHg?e=8r1SUL&download=1](https://epcbocc.sharepoint.com/:x:/s/Share/EYXZ5t16UJFGk1rzIU91VogBa8U37lh8z_Hftf2KJISSHg?e=8r1SUL&download=1). The files can be viewed at [https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv\\_6-t7kBUYaXiIqN0B1n2w57Z\\_V3kQ?e=NdZQcU](https://epcbocc.sharepoint.com/:f:/s/Share/EiypSSYdsEFCi84Sv_6-t7kBUYaXiIqN0B1n2w57Z_V3kQ?e=NdZQcU).

Water quality parameters returned by default are total nitrogen (tn), Secchi depth (sd), chlorophyll-a (chl<sub>a</sub>), salinity (top, mid, and bottom depths, Sal\_ prefix), water temperature (top, mid, and bottom depths, Temp\_Water\_ prefix), turbidity (Turbidity\_JTU-NTU), and water color (Color\_345\_F45). Additional qualifier columns for each that include the \_Q suffix are also returned, excluding salinity and water temperature. All other water quality parameters and qualifiers can be returned by setting all = T.

**Value**

A data frame of formatted water quality data.

**See Also**

[read\\_formwq](#), [read\\_importphyto](#)

**Examples**

```
## Not run:
# file path
xlsx <- '~/Desktop/RWMDDataSpreadsheet_ThroughCurrentReportMonth.xlsx'

# load and assign to object
epcdata <- read_importwq(xlsx, download_latest = T)

# get all water quality parameters
epcdataall <- read_importwq(xlsx, download_latest = T, all = T)

## End(Not run)
```

---

read_importwqp	<i>Import data from the Water Quality Portal</i>
----------------	--

---

### Description

Import data from the Water Quality Portal

### Usage

```
read_importwqp(org, type, trace = F)
```

### Arguments

org	chr string indicating the organization identifier, see details
type	chr string indicating data type to download, one of "wq" or "fib"
trace	logical indicating whether to display progress messages, default FALSE

### Details

This function retrieves data from the Water Quality Portal API (<https://www.waterqualitydata.us/>) for selected counties in or around the Tampa Bay watershed. The type of data returned are defined by the type argument as either "wq" or "fib" for water quality of Fecal Indicator Bacteria data, respectively.

The org argument retrieves data for the specific organization. Valid entries for org include "21FLCOSP\_WQX" (City of St. Petersburg), "21FLDOH\_WQX" (Florida Department of Health), "21FLHILL\_WQX" (Hillsborough County), "21FLMANA\_WQX" (Manatee County), "21FLPASC\_WQX" (Pasco County), "21FLPDEM\_WQX" (Pinellas County), "21FLPOLK\_WQX" (Polk County), or "21FLTPA\_WQX" (Florida Department of Environmental Protection, Southwest District). The naming convention follows the Organization ID in the Water Quality Portal.

The function fetches results and station metadata, combines and formats them using the read\_formwqp function, and returns the processed data as a data frame. Parameters are specific to the type argument.

### Value

A data frame containing the imported data for the selected county

### See Also

[read\\_formwqp](#)

## Examples

```
## Not run:  
# get Manatee County water quality data  
mancodata <- read_importwqp(org = '21FLMANA_WQX', type = 'wq', trace = T)  
  
# get Pinellas County FIB data  
pincodata <- read_importwqp(org = '21FLPDEM_WQX', type = 'fib', trace = T)  
  
## End(Not run)
```

---

read_transect	<i>Import JSON seagrass transect data from Water Atlas</i>
---------------	--

---

## Description

Import JSON seagrass transect data from Water Atlas

## Usage

```
read_transect(training = FALSE, raw = FALSE)
```

## Arguments

training	logical if training data are imported or the complete database
raw	logical indicating if raw, unformatted data are returned, see details

## Details

The function imports a JSON file from the USF Water Atlas. If `training = TRUE`, a dataset from the TBEP training survey is imported from <http://dev.seagrass.wateratlas.usf.edu/api/assessments/training>. If `training = FALSE`, the entire transect survey database is imported from [http://dev.seagrass.wateratlas.usf.edu/api/assessments/all\\_\\_use-with-care](http://dev.seagrass.wateratlas.usf.edu/api/assessments/all__use-with-care).

Abundance is reported as a numeric value from 0 -5 for Braun-Blanquet coverage estimates, blade length is in cm, and short shoot density is number of shoots per square meter. The short density is corrected for quadrat size included in the raw data.

If `raw = TRUE`, the unformatted data are returned. The default is to use formatting that allows the raw data to be used with the downstream functions. The raw data may have extra information that may be of use outside of the plotting functions in this package.

## Value

data frame

**Examples**

```
## Not run:
# get training data
transect <- read_transect(training = TRUE)

# import all transect data
transect <- read_transect()

## End(Not run)
```

---

seagrass	<i>Seagrass coverage by year</i>
----------	----------------------------------

---

**Description**

Seagrass coverage by year

**Usage**

```
seagrass
```

**Format**

A data frame used to create the flagship seagrass coverage graphic:

**Year** int  
**Acres** num  
**Hectares** num

**Details**

Original data are from the Southwest Florida Water Management District and available online at <https://data-swfwmd.opendata.arcgis.com/>. Totals are for all of Tampa Bay.

**Examples**

```
## Not run:

seagrass <- structure(list(
  Year = c(1950L, 1982L, 1988L, 1990L, 1992L, 1994L, 1996L,
    1999L, 2001L, 2004L, 2006L, 2008L, 2010L, 2012L, 2014L,
    2016L, 2018L, 2020L, 2022L),
  Acres = c(40420, 21650, 23285, 25226, 25753, 26518, 26916,
    24841, 26078, 27021, 28299, 29647, 32897, 34642, 40294.71,
    41655.16, 40651.55, 34298, 30137),
  Hectares = c(16357.39, 8761.44, 9423.11, 10208.6, 10421.87,
    10731.45, 10892.52, 10052.8, 10553.39, 10935.01, 11452.2,
    11997.72, 13312.94, 14019.27, 16306.69, 16857.25, 16451.1,
```

```
      13880, 12196)
    ), class = "data.frame", row.names = c(NA, -19L))

save(seagrass, file = 'data/seagrass.RData', compress = 'xz')

## End(Not run)
```

---

sedimentdata

*Sediment data for the Tampa Bay current as of 12142023*

---

### Description

Sediment data for the Tampa Bay current as of 12142023

### Usage

```
sedimentdata
```

### Format

A data.frame with 216627 rows and 24 variables:

**ProgramId** int  
**ProgramName** chr  
**FundingProject** chr  
**yr** int  
**AreaAbbr** chr  
**StationID** int  
**StationNumber** chr  
**Latitude** num  
**Longitude** num  
**Replicate** chr  
**SedResultsType** chr  
**Parameter** chr  
**ValueAdjusted** num  
**Units** chr  
**Qualifier** chr  
**MDLnum** num  
**PQLnum** num  
**TEL** num  
**PEL** num

**BetweenTELPEL** chr  
**ExceedsPEL** chr  
**PreparationDate** chr  
**AnalysisTimeMerge** chr  
**PELGrade** Factor

### Examples

```

## Not run:
# location to download data
path <- '~/Desktop/sediment.zip'

# load and assign to object
sedimentdata <- read_importsediment(path, download_latest = TRUE, remove = TRUE)

save(sedimentdata, file = 'data/sedimentdata.RData', compress = 'xz')

## End(Not run)

```

---

 sgmanagement

*Seagrass management areas for Tampa Bay*


---

### Description

Seagrass management areas for Tampa Bay

### Usage

```
sgmanagement
```

### Format

A simple features `sf` object (MULTIPOLYGON) with 30 features and 1 field, `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`

**areas** int

### Details

These polygons are seagrass management areas for Tampa Bay that provide a finer division of areas within major segments (`tbseg`) having relevance for locations of seagrass beds.



**Examples**

```

## Not run:
library(sf)
library(dplyr)
library(tools)

# NAD83(HARN) / Florida West (ftUS)
# same as sgseg
prj <- 2882

# create sf object of boundaries
sgmanagement <- st_read(
  dsn = '~/Desktop/TBEP/GISboundaries/Seagrass_Management_Areas/TBEP_SG_MA_FINAL_Projectfix.shp',
  drivers = 'ESRI Shapefile'
) %>%
  select(areas = TBEP_SG_MA) %>%
  st_zm() %>%
  st_transform(prj)

# save
save(sgmanagement, file = 'data/sgmanagement.RData', compress = 'xz')

## End(Not run)

```

---

sgseg

*Seagrass segment reporting boundaries for southwest Florida*


---

**Description**

Seagrass segment reporting boundaries for southwest Florida

**Usage**

```
sgseg
```

**Format**

A simple features `sf` object (POLYGON) with 22 features and 1 field, `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`

**segment** chr

**Details**

These polygons are used by Southwest Florida Water Management District for summarizing sea-grass coverage estimates by major coastal and estuarine boundaries.

**Examples**

```

## Not run:
library(sf)
library(dplyr)
library(tools)

# create sf object of boundaries
# make sure projection does not change
sgseg <- st_read(
  dsn = '~/Desktop/TBEP/GISboundaries/Seagrass_Segment_Boundaries/Seagrass_Segment_Boundaries.shp',
  drivers = 'ESRI Shapefile'
) %>%
select(segment = SEAGRASSSE) %>%
mutate(
  segment = tolower(segment),
  segment = case_when(
    segment == 'terra ciea bay' ~ 'Terra Ceia Bay',
    T ~ segment
  ),
  segment = toTitleCase(segment)
)

# save
save(sgseg, file = 'data/sgseg.RData', compress = 'xz')

## End(Not run)

```

---

show\_annualassess

---

*Create a table for the annual management outcome assessments*


---

**Description**

Create a table for the annual management outcome assessments for chlorophyll-a and light attenuation by bay segment

**Usage**

```

show_annualassess(
  epccdata,
  yrsel,
  partialyr = F,
  caption = F,
  family = "Arial",
  txtsz = 12,
  width = NULL
)

```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importwq</a>
yrsel	numeric indicating chosen year
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
caption	logical indicating if a caption is added using <a href="#">set_caption</a>
family	chr string indicating font family for text labels
txtsz	numeric indicating font size
width	optional numeric value indicating width in inches

**Value**

A [flextable](#) object showing the segment-averaged chlorophyll-a and light attenuation for the selected year, with bay segment names colored by the management outcome used in [show\\_matrix](#).

**Examples**

```
show_annualassess(epcdata, yrsel = 2023)
show_annualassess(epcdata, yrsel = 2023, caption = TRUE)
```

---

show_boxplot	<i>Plot monthly chlorophyll or light attenuation values for a segment</i>
--------------	---

---

**Description**

Plot monthly chlorophyll or light attenuation values for a bay segment

**Usage**

```
show_boxplot(
  epcdata,
  param = c("chla", "la"),
  yrsel = NULL,
  yrrng = c(1975, 2023),
  ptsz = 0.5,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  trgs = NULL,
  family = NA,
  labelexp = TRUE,
  txtlab = TRUE,
  partialyr = FALSE
)
```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importwq</a>
param	chr string for which parameter to plot, one of "chl <sub>a</sub> " for chlorophyll or "l <sub>a</sub> " for light attenuation
yrsel	numeric for year to emphasize, shown as separate red points on the plot
yrrng	numeric vector indicating min, max years to include
ptsz	numeric indicating point size of observations not in yrsel
bay_segment	chr string for the bay segment, one of "OTB", "HB", "MTB", "LTB"
trgs	optional data.frame for annual bay segment water quality targets, defaults to <a href="#">targets</a>
family	optional chr string indicating font family for text labels
labelexp	logical indicating if y axis and target labels are plotted as expressions, default TRUE
txtlab	logical indicating if a text label for the target value is shown in the plot
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter

**Details**

Points not included in yrsel are plotted over the box plots using [position\\_jitter](#). Use ptsz = -1 to suppress. The dotted line in the plot shows the large exceedance value.

**Value**

A [ggplot](#) object

**Examples**

```
show_boxplot(epcdata, bay_segment = 'OTB')
```

---

show\_compplot

*Make a bar plot for transect training group comparisons*

---

**Description**

Make a bar plot for transect training group comparisons

**Usage**

```
show_complot(  
  transect,  
  yr,  
  site,  
  species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia"),  
  varplo = c("Abundance", "Blade Length", "Short Shoot Density"),  
  base_size = 18,  
  txtxt = 10,  
  size = 1  
)
```

**Arguments**

transect	data frame returned by <a href="#">read_transect</a>
yr	numeric for year of training data to plot
site	chr string indicating site results to plot
species	chr string indicating which species to plot
varplo	chr string indicating which variable to plot
base_size	numeric indicating text scaling size for plot
txtxt	numeric indicating text size for x-axis labels
size	numeric indicating line size

**Value**

A [ggplot](#) object

**Examples**

```
transect <- read_transect(training = TRUE)  
show_complot(transect, yr = 2023, site = '2', species = 'Halodule', varplo = 'Abundance')
```

---

show\_enteromap

*Map Enterococcus results by month, year, and location*

---

**Description**

Map Enterococcus results by month, year, and location

**Usage**

```
show_enteromap(
  fibdata,
  yrsel,
  mosel,
  areasel = NULL,
  wetdry = FALSE,
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL
)
```

**Arguments**

fibdata	data frame of Enterococcus sample data as returned by <a href="#">enterodata</a> or <a href="#">anlz_fibwetdry</a>
yrsel	optional numeric to filter data by year
mosel	optional numeric to filter data by month
areasel	optional character string to filter output by stations in the long_name column of enterodata, see details
wetdry	logical; if TRUE, incorporate wet/dry differences (this will result in a call to <a href="#">anlz_fibwetdry</a> , in which case temporal_window and wet_threshold are required). If FALSE (default), do not differentiate between wet and dry samples.
precipdata	input data frame as returned by <a href="#">read_importrain</a> . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object <a href="#">catchprecip</a> has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to <a href="#">catchprecip</a> .
temporal_window	numeric; required if wetdry is TRUE. number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric; required if wetdry is TRUE. inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period

**Details**

Placing the mouse cursor over an item on the map will reveal additional information about a station.

**Value**

A leaflet map for the selected year, month, and area showing stations and FIB concentration category

**See Also**

[anlz\\_enteromap](#) for details on the categories

**Examples**

```

show_enteromap(enterodata, yrsel = 2020, mosel = 9)

# wet/dry samples
show_enteromap(enterodata, yrsel = 2020, mosel = 9, wetdry = TRUE,
               temporal_window = 2, wet_threshold = 0.5)

# Old Tampa Bay only
show_enteromap(enterodata, yrsel = 2020, mosel = 9, areasel = "Old Tampa Bay")

```

---

show_fibmap	<i>Map Fecal Indicator Bacteria (FIB) results by month, year, and location</i>
-------------	--

---

**Description**

Map Fecal Indicator Bacteria (FIB) results by month, year, and location

**Usage**

```
show_fibmap(fibdata, yrsel, mosel, areasel = NULL)
```

**Arguments**

fibdata	input FIB data.frame as returned by <a href="#">read_importfib</a>
yrsel	optional numeric value to filter output by years in fibdata
mosel	optional numeric value to filter output by month in fibdata
areasel	optional character string to filter output by stations in the area column of fibdata, see details

**Details**

Placing the mouse cursor over an item on the map will reveal additional information about a station.

**Value**

A leaflet map for the selected year, month, and area showing stations and FIB concentration category

**See Also**

[anz\\_fibmap](#) for details on the categories

**Examples**

```
# show selected year, month
show_fibmap(fibdata, yrsel = 2023, mosel = 8)

# show selected year, month, and area
show_fibmap(fibdata, yrsel = 2020, mosel = 7, areasel = 'Alafia River')
```

---

show\_fibmatmap      *Map Fecal Indicator Bacteria matrix results by year*

---

**Description**

Map Fecal Indicator Bacteria matrix results by year

**Usage**

```
show_fibmatmap(
  fibdata,
  yrsel,
  areasel,
  indic,
  threshold = NULL,
  lagyr = 3,
  subset_wetdry = c("all", "wet", "dry"),
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  listout = FALSE,
  warn = TRUE
)
```

**Arguments**

fibdata	input data frame as returned by <a href="#">read_importfib</a> or <a href="#">read_importentero</a>
yrsel	numeric value indicating the year to map
areasel	vector of bay segment or area names to include, see details
indic	character for choice of fecal indicator. Allowable options are fcolif for fecal coliform, or entero for Enterococcus. A numeric column in the data frame must have this name.
threshold	optional numeric for threshold against which to calculate exceedances for the indicator bacteria of choice. If not provided, defaults to 400 for fcolif and 130 for entero.
lagyr	numeric for year lag to calculate categories, see details



subset_wetdry	character, subset data frame to only wet or dry samples as defined by wet_threshold and temporal_window? Defaults to "all", which will not subset. If "wet" or "dry" is specified, <a href="#">anlz_fibwetdry</a> is called using the further specified parameters, and the data frame is subsetted accordingly.
precipdata	input data frame as returned by <a href="#">read_importtrain</a> . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object <a href="#">catchprecip</a> has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to <a href="#">catchprecip</a> .
temporal_window	numeric; required if subset_wetdry is not "all". number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric; required if subset_wetdry is not "all". inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
listout	logical to return a list of simple feature objects for the data used in the leaflet map, default FALSE
warn	logical to print warnings about stations with insufficient data, default TRUE

### Details

Placing the mouse cursor over an item on the map will reveal additional information about a segment or station.

If the input to fibdata is from EPCHC (from [read\\_importfib](#)), valid entries for areasel include 'Alafia River', 'Hillsborough River', 'Big Bend', 'Cockroach Bay', 'East Lake Outfall', 'Hillsborough Bay', 'Little Manatee River', 'Lower Tampa Bay', 'McKay Bay', 'Middle Tampa Bay', 'Old Tampa Bay', 'Palm River', 'Tampa Bypass Canal', and 'Valrico Lake'. If the input data is not from EPCHC (from [read\\_importentero](#)), valid entries for areasel include 'OTB', 'HB', 'MTB', 'LTB', 'BCB', and 'MR'.

Bay segment matrix categories are shown in addition to stations if fibsel input is not from EPCHC (from [read\\_importentero](#)). Stations for these data were chosen specifically as downstream endpoints for each bay segment, whereas the EPCHC data are not appropriate for estimating bay segment outcomes.

### Value

A leaflet map for the selected year and area showing station matrix scores if listout = FALSE (default). Bay segment scores are also shown if the input data are not from EPCHC. A list of simple feature objects is returned if listout = TRUE.

### See Also

[anlz\\_fibmatrix](#) for details on the categories

### Examples

```
# non-EPCHC data
show_fibmatmap(enterodata, yrsl = 2020, indic = 'entero', areasel = 'OTB')
```

```
# EPCHC data
show_fibmatmap(fibdata, yrsl = 2016, indic = 'fcolif',
  areasel = c("Hillsborough River", "Alafia River"))
```

---

show_fibmatrix	<i>Plot a matrix of Fecal Indicator Bacteria categories over time by station or bay segment</i>
----------------	---

---

## Description

Plot a matrix of Fecal Indicator Bacteria categories over time by station or bay segment

## Usage

```
show_fibmatrix(
  fibdata,
  yrrng = NULL,
  stas = NULL,
  bay_segment = NULL,
  indic = NULL,
  threshold = NULL,
  lagyr = 3,
  subset_wetdry = c("all", "wet", "dry"),
  precipdata = NULL,
  temporal_window = NULL,
  wet_threshold = NULL,
  txtsz = 3,
  asreact = FALSE,
  nrows = 10,
  family = NA,
  angle = 90,
  size = 10,
  hjust = 0,
  plotly = FALSE,
  width = NULL,
  height = NULL,
  warn = TRUE
)
```

## Arguments

fibdata	input data frame as returned by <a href="#">read_importfib</a> or <a href="#">read_importentero</a>
yrrng	numeric vector indicating min, max years to include, defaults to range of years in data, see details
stas	optional vector of stations to include, see details
bay_segment	optional vector of bay segment names to include, supercedes stas if provided, see details

indic	character for choice of fecal indicator. Allowable options are fcolif for fecal coliform, or entero for Enterococcus. A numeric column in the data frame must have this name.
threshold	optional numeric for threshold against which to calculate exceedances for the indicator bacteria of choice. If not provided, defaults to 400 for fcolif and 130 for entero.
lagyr	numeric for year lag to calculate categories, see details
subset_wetdry	character, subset data frame to only wet or dry samples as defined by wet_threshold and temporal_window? Defaults to "all", which will not subset. If "wet" or "dry" is specified, <a href="#">anlz_fibwetdry</a> is called using the further specified parameters, and the data frame is subsetted accordingly.
precipdata	input data frame as returned by <a href="#">read_importtrain</a> . columns should be: station, date (yyyy-mm-dd), rain (in inches). The object <a href="#">catchprecip</a> has this data from 1995-2023 for select Enterococcus stations. If NULL, defaults to <a href="#">catchprecip</a> .
temporal_window	numeric; required if subset_wetdry is not "all". number of days precipitation should be summed over (1 = day of sample only; 2 = day of sample + day before; etc.)
wet_threshold	numeric; required if subset_wetdry is not "all". inches accumulated through the defined temporal window, above which a sample should be defined as being from a 'wet' time period
txtsz	numeric for size of text in the plot, applies only if tab = FALSE. Use txtsz = NULL to suppress.
asreact	logical indicating if a <a href="#">reactable</a> object is returned
nrows	if asreact = TRUE, a numeric specifying number of rows in the table
family	optional chr string indicating font family for text labels
angle	numeric for angle of x-axis text labels
size	numeric for size of the x-axis text labels
hjust	numeric for horizontal justification of x-axis text labels
plotly	logical if matrix is created using plotly
width	numeric for width of the plot in pixels, only applies if plotly = TRUE
height	numeric for height of the plot in pixels, only applies if plotly = TRUE
warn	logical to print warnings about stations with insufficient data, default TRUE

## Details

The matrix color codes years and stations based on the likelihood of fecal indicator bacteria concentrations exceeding 400 CFU / 100 mL for Fecal Coliform or 130 CFU / 100 mL for Enterococcus. Bay segments are used instead of stations if bay\_segment is not NULL and the input data are from [read\\_importentero](#). The likelihoods are categorized as A, B, C, D, or E (Microbial Water Quality Assessment or MWQA categories) with corresponding colors, where the breakpoints for each category are <10%, 10-30%, 30-50%, 50-75%, and >75% (right-closed). By default, the results for each year are based on a right-centered window that uses the previous two years and the current year

to calculate probabilities from the monthly samples (lagyr = 3). Methods and rationale for this categorization scheme are provided by the Florida Department of Environmental Protection, Figure 8 in the document at <http://publicfiles.dep.state.fl.us/DEAR/BMAP/Tampa/MST%20Report/Fecal%20BMAP%20DST%20Final%20Report%20--%20June%202008.pdf> and Morrison et al. 2009 in the **BASIS 5** proceedings.

The default stations for fecal coliform data are those used in TBEP report #05-13 (<https://drive.google.com/file/d/1MZnK3cMzV7LRg6dTbCKX8AOZU0GNurJJ/view>) for the Hillsborough River Basin Management Action Plan (BMAP) subbasins if bay\_segment is NULL and the input data are from `read_importfib`. These include Blackwater Creek (WBID 1482, EPC stations 143, 108), Baker Creek (WBID 1522C, EPC station 107), Lake Thonotosassa (WBID 1522B, EPC stations 135, 118), Flint Creek (WBID 1522A, EPC station 148), and the Lower Hillsborough River (WBID 1443E, EPC stations 105, 152, 137). Other stations can be plotted using the `stas` argument.

### Value

A static `ggplot` object is returned by default. A `reactable` table is returned if `asreact = TRUE`. An interactive `plotly` object is returned if `plotly = TRUE`.

### Examples

```
show_fibmatrix(fibdata)

# change the threshold
show_fibmatrix(fibdata, threshold = 200)

# change the indicator
show_fibmatrix(fibdata, indic = 'entero')

# show matrix for only dry samples
show_fibmatrix(enteredata, indic = 'entero', lagyr = 1, subset_wetdry = "dry",
               temporal_window = 2, wet_threshold = 0.5)
```

---

show\_hmpreport

*Show Habitat Master Plan progress report card*

---

### Description

Show Habitat Master Plan progress report card

### Usage

```
show_hmpreport(
  acres,
  subtacres,
  hmptrgs,
  typ,
  twocol = FALSE,
  strata = c("Subtidal", "Intertidal", "Supratidal"),
```

```

    ycollapse = FALSE,
    text = 2.5,
    xang = 25,
    family = NA,
    width = NULL,
    height = NULL
  )

```

### Arguments

acres	data.frame for intertidal and supratidal land use and cover of habitat types for each year of data
subt acres	data.frame for subtidal cover of habitat types for each year of data
hmptrgs	data.frame of Habitat Master Plan targets and goals
typ	character string indicating "targets" or "goals"
twocol	logical indicating only two colors are shown if target or goals are met and symbols indicate the likelihood of attaining targets or goals, see details
strata	character string indicating with strata to plot, one to many of "Subtidal", "Intertidal", and "Supratidal"
ycollapse	logical indicating if the y-axis is collapsed to year with data, see details
text	numeric indicating text size for proportion of target or goal met for habitat types shown in each cell types or symbols if twocol = T, use NULL to suppress
xang	numeric for angle of habitat labels on the x-axis (top)
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

### Details

If `twocol = F`, colors indicate both if the target/goal is met and the likelihood of attaining the target/goal by 2030/2050. Red indicates the target/goal is not met and will likely not be met by 2030/2050 (trending below target/goal), yellow indicates the target/goal is met although it likely will not be met by 2030/2050 (trending below target/goal), light green indicates the target/goal is not met although it will likely be met by 2030/2050 (trending above target/goal), and green indicates the target/goal is met and will likely be met by 2030/2050 (trending above target/goal). Numbers in each cell show the proportion of the target or goal met at each year where data are available. If `twocol = T`, the colors indicate if the goal is met (green) or not met (red) and the symbols in each cell indicate if the goal is likely to be met (up arrow) or not (down arrow) by 2030/2050. In both cases, the colors and trends are relative to the 2030 targets or 2050 goals using the `typ` argument.

The report card provides no information on artificial reefs, living shorelines, and hard bottom habitats. These habitats are not assessed in routine data products from the Southwest Florida Water Management District, although targets and goals are provided in the Habitat Master Plan.

The subtidal data in `subt acres` and the inter/supratidal data in `acres` are provided as different datasets by the Southwest Florida Water Management District. The years in each dataset typically do not match and each dataset is collected at approximate 2 to 3 year intervals. By default, year on the y-axis is shown as a continuous variable, where gaps are shown in years when each dataset was unavailable. Use `ycollapse = TRUE` to remove years without data.

**Value**

A [ggplot2](#) object showing overall progress in attaining Habitat Master Plan targets or goals.

**Examples**

```
# view summarized data for report card, targets
show_hmpreport(acres, subtacres, hmptrgs, typ = "targets")

# view summarized data for report card, goals
show_hmpreport(acres, subtacres, hmptrgs, typ = "goals")

# remove empty years
show_hmpreport(acres, subtacres, hmptrgs, typ = "targets", ycollapse = TRUE)

# select only subtidal
show_hmpreport(acres, subtacres, hmptrgs, typ = "targets", ycollapse = TRUE, strata = 'Subtidal')
```

---

 show\_matrix

---

*Create a colorized table for indicator reporting*


---

**Description**

Create a colorized table for indicator reporting

**Usage**

```
show_matrix(
  epcdata,
  txtsz = 3,
  trgs = NULL,
  yrrng = NULL,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  asreact = FALSE,
  nrows = 10,
  abbrev = FALSE,
  family = NA,
  historic = TRUE,
  plotly = FALSE,
  partialyr = FALSE,
  width = NULL,
  height = NULL
)
```

**Arguments**

epcdata            data frame of epc data returned by [read\\_importtwq](#)  
 txtsz             numeric for size of text in the plot, applies only if tab = FALSE

trgs	optional data.frame for annual bay segment water quality targets, defaults to <a href="#">targets</a>
yrrng	numeric vector indicating min, max years to include, defaults to range of years in epccdata
bay_segment	chr string for bay segments to include, one to all of "OTB", "HB", "MTB", "LTB"
asreact	logical indicating if a <a href="#">reactable</a> object is returned
nrows	if asreact = TRUE, a numeric specifying number of rows in the table
abbrev	logical indicating if text labels in the plot are abbreviated as the first letter
family	optional chr string indicating font family for text labels
historic	logical if historic data are used from 2005 and earlier
plotly	logical if matrix is created using plotly
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

**Value**

A static [ggplot](#) object is returned if asreact = FALSE, otherwise a [reactable](#) table is returned

**See Also**

[show\\_wqmatrix](#), [show\\_segmatrix](#)

**Examples**

```
show_matrix(epccdata)
```

---

show\_matrixplotly      *Creates a plotly matrix from any matrix function input*

---

**Description**

Creates a plotly matrix from any matrix function input

**Usage**

```
show_matrixplotly(
  mat,
  family = NA,
  tooltip = "Result",
  width = NULL,
  height = NULL
)
```

**Arguments**

mat	input matrix as output from <a href="#">show_matrix</a> , <a href="#">show_segmatrix</a> , <a href="#">show_wqmatrix</a> , or <a href="#">show_tbnimatrix</a>
family	optional chr string indicating font family for text labels
tooltip	chr string indicating the column name for tooltip
width	numeric for width of the plot in pixels
height	numeric for height of the plot in pixels

**Value**

A [plotly](#) data object

**Examples**

```
mat <- show_wqmatrix(epcdata)
show_matrixplotly(mat)
```

---

show_ratab	<i>Create a bay segment assessment table for the 2022-2026 reasonable assurance period</i>
------------	--

---

**Description**

Create a bay segment assessment table for the 2022-2026 reasonable assurance period

**Usage**

```
show_ratab(
  epcdata,
  yrsel,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "RALTB"),
  partialyr = F,
  outtxt1 = NULL,
  outtxt2 = NULL,
  outtxt3 = NULL,
  outtxt45 = NULL,
  txtsz = 13,
  width = NULL
)
```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importtwq</a>
yrsel	numeric indicating chosen year
bay_segment	chr string for the bay segment, one of "OTB", "HB", "MTB", "LTB", "RALTB"



partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
outtxt1	optional text for NMC action 1, added to the outcome column
outtxt2	optional text for NMC action 2, added to the outcome column
outtxt3	optional text for NMC action 3, added to the outcome column
outtxt45	optional text for NMC actions 4 and 5, added to the outcome column
txtsz	numeric indicating font size
width	optional numeric value indicating width in inches

### Details

Choosing bay\_segment = 'RALTB' will not work with [epcdata](#) and additional data are needed to use this option.

### Value

A [flextable](#) object showing the reasonable assurance compliance of the bay segment for the selected year within the five-year period.

### Examples

```
show_ratab(epcdata, yrse1 = 2023, bay_segment = 'OTB')
```

---

show_reactable	<i>Create reactable table from matrix data</i>
----------------	--

---

### Description

Create reactable table from matrix data

### Usage

```
show_reactable(totab, colfun, nrows = 10, txtsz = 3)
```

### Arguments

totab	A data frame in wide format of summarized results
colfun	Function specifying how colors are treated in cell background
nrows	numeric specifying number of rows in the table
txtsz	numeric indicating text size in the cells, use txtsz = NULL to suppress

### Details

This function is used internally within [show\\_matrix](#) and [show\\_wqmatrix](#)

**Value**

A [reactable](#) table

**Examples**

```

data(targets)
data(epcdata)

library(tidyr)
library(dplyr)

# data
totab <- anlz_avedat(epcdata) %>%
  .$ann %>%
  filter(var %in% 'mean_chla') %>%
  left_join(targets, by = 'bay_segment') %>%
  select(bay_segment, yr, val, chla_thresh) %>%
  mutate(
    bay_segment = factor(bay_segment, levels = c('OTB', 'HB', 'MTB', 'LTB')),
    outcome = case_when(
      val < chla_thresh ~ 'green',
      val >= chla_thresh ~ 'red'
    )
  ) %>%
  select(bay_segment, yr, outcome) %>%
  spread(bay_segment, outcome)

# color function
colfun <- function(x){

  out <- case_when(
    x == 'red' ~ '#FF3333',
    x == 'green' ~ '#33FF3B'
  )

  return(out)
}

show_reactable(totab, colfun)

```

---

show\_seagrasscoverage *Create a barplot of seagrass coverage over time in Tampa Bay*

---

**Description**

Create a barplot of seagrass coverage over time in Tampa Bay

**Usage**

```
show_seagrasscoverage(  
  seagrass,  
  maxyr = 2022,  
  family = NA,  
  lastlab = "acres",  
  axsbrk = c(0.08, 0.1)  
)
```

**Arguments**

seagrass	input data.frame included with the package as <a href="#">seagrass</a>
maxyr	numeric for maximum year to plot
family	optional chr string indicating font family for text labels
lastlab	chr string of text to append to label in last bar of the barplot
axsbrk	numeric vector of length two indicating where the x-axis break occurs

**Details**

This function creates the flagship seagrass coverage graphic to report on coverage changes over time. All data were pre-processed and included in the package as the [seagrass](#) dataset. Original data are from the Southwest Florida Water Management District and available online at <https://data-swfwmf.opendata.arcgis.com/>. This function and the data used to create the plot are distinct from those used for the transect monitoring program.

**Value**

A grid object showing acres of total seagrass coverage in Tampa Bay over time.

**Examples**

```
show_seagrasscoverage(seagrass)
```

---

show\_sedimentalratio *Plot metal concentrations against aluminum*

---

**Description**

Plot metal concentrations against aluminum

**Usage**

```
show_sedimentalratio(
  sedimentdata,
  param,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  lnsz = 1,
  base_size = 12,
  plotly = FALSE,
  family = NA,
  width = NULL,
  height = NULL
)
```

**Arguments**

sedimentdata	input sediment data.frame as returned by <a href="#">read_importsediment</a>
param	chr string for which parameter to plot, must be a metal
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
lnsz	numeric for line size
base_size	numeric indicating text scaling size for plot
plotly	logical if matrix is created using plotly
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

**Details**

The plot shows the selected contaminant concentration relative to aluminum, the latter being present as a common metal in the Earth's crust. An elevated ratio of a metal parameter relative to aluminum suggests it is higher than background concentrations.

Lines for the Threshold Effect Level (TEL) and Potential Effect Level (PEL) are shown for the parameter, if available.

**Value**

A [ggplot](#) object or a [plotly](#) object if plotly = TRUE showing the ratio of the selected parameter plotted against aluminum concentrations collected at the same site. Black lines show the linear fit of a log-log model and the 95

## References

Schropp, S. J., Graham Lewis, F., Windom, H. L., Ryan, J. D., Calder, F. D., & Burney, L. C. 1990. Interpretation of metal concentrations in estuarine sediments of Florida using aluminum as a reference element. *Estuaries*. 13:227-235.

## Examples

```
show_sedimentalratio(sedimentdata, param = 'Arsenic')
```

---

show\_sedimentave      *Plot sediment concentration averages by bay segment*

---

## Description

Plot sediment concentration averages by bay segment

## Usage

```
show_sedimentave(
  sedimentdata,
  param,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  lnsz = 1,
  base_size = 12,
  plotly = FALSE,
  family = NA,
  width = NULL,
  height = NULL
)
```

## Arguments

sedimentdata	input sediment data.frame as returned by <a href="#">read_importsediment</a>
param	chr string for which parameter to plot
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
lnsz	numeric for line size
base_size	numeric indicating text scaling size for plot
plotly	logical if matrix is created using plotly

family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

### Details

Lines for the Threshold Effect Level (TEL) and Potential Effect Level (PEL) are shown for the parameter, if available. Confidence intervals may not be shown for segments with insufficient data.

### Value

A `ggplot` object or a `plotly` object if `plotly = TRUE` showing sediment averages and 95

### Examples

```
show_sedimentave(sedimentdata, param = 'Arsenic')
```

---

show_sedimentmap	<i>Make a map for sediment contaminants at stations in Tampa Bay</i>
------------------	--

---

### Description

Make a map for sediment contaminants at stations in Tampa Bay

### Usage

```
show_sedimentmap(
  sedimentdata,
  param,
  yrrng = c(1993, 2022),
  funding_proj = "TBEP",
  weight = 1.5
)
```

### Arguments

sedimentdata	input sediment data.frame as returned by <code>read_importsediment</code>
param	chr string for which parameter to plot
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
weight	numeric for outline width of station points on the map

**Details**

The map shows sediment contaminant concentrations for the selected parameter relative to Threshold Effects Levels (TEL) and Potential Effects Levels (PEL), if available. Green points show concentrations below the TEL, yellow points show concentrations between the TEL and PEL, and red points show concentrations above the PEL. The applicable TEL and PEL values for the parameter are indicated in the legend. If TEL and PEL thresholds are not available, a map of the sediment concentrations is shown and a warning is returned to the console.

**Value**

A [leaflet](#) object

**Examples**

```
show_sedimentmap(sedimentdata, param = 'Arsenic')
```

---

show\_sedimentpelave *Plot summary of PEL averages by bay segment*

---

**Description**

Plot summary of PEL averages by bay segment

**Usage**

```
show_sedimentpelave(
  sedimentdata,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  alph = 1,
  ylim = c(0, 0.4),
  lnsz = 1,
  base_size = 12,
  plotly = FALSE,
  family = NA,
  width = NULL,
  height = NULL
)
```

**Arguments**

sedimentdata	input sediment data.frame as returned by <a href="#">read_importsediment</a>
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"

funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
alph	numeric indicating alpha value for score category colors
ylim	numeric for y axis limits
lnsz	numeric for line size
base_size	numeric indicating text scaling size for plot
plotly	logical if matrix is created using plotly
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

### Details

Lines for the grades are shown on the plot. Confidence intervals may not be shown for segments with insufficient data.

### Value

A [ggplot](#) object or a [plotly](#) object if plotly = TRUE showing PEL averages (average of the averages) and 95

### Examples

```
show_sedimentpelave(sedimentdata)
```

---

```
show_sedimentpelaveplotly
```

*Creates a plotly object for average PEL plots*

---

### Description

Creates a plotly object for average PEL plots

### Usage

```
show_sedimentpelaveplotly(p, width = NULL, height = NULL)
```

### Arguments

p	<a href="#">ggplot</a> object as output from <a href="#">show_sedimentpelave</a>
width	numeric for width of the plot in pixels
height	numeric for height of the plot in pixels

### Value

A [plotly](#) data object



## Examples

```
p <- show_sedimentpelave(sedimentdata)
show_sedimentpelaveplotly(p)
```

---

```
show_sedimentpelmap Make a map for sediment PEL ratios at stations in Tampa Bay
```

---

## Description

Make a map for sediment PEL ratios at stations in Tampa Bay

## Usage

```
show_sedimentpelmap(
  sedimentdata,
  yrrng = c(1993, 2022),
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"),
  funding_proj = "TBEP",
  weight = 1.5
)
```

## Arguments

sedimentdata	input sediment data.frame as returned by <a href="#">read_importsediment</a>
yrrng	numeric vector indicating min, max years to include, use single year for one year of data
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
funding_proj	chr string for the funding project, one to many of "TBEP" (default), "TBEP-Special", "Apollo Beach", "Janicki Contract", "Rivers", "Tidal Streams"
weight	numeric for outline width of station points on the map

## Details

The map shows average PEL ratios graded from A to F for benthic stations monitored in Tampa Bay. The PEL is a measure of how likely a contaminant is to have a toxic effect on invertebrates that inhabit the sediment. The PEL ratio is the contaminant concentration divided by the Potential Effects Levels (PEL) that applies to a contaminant, if available. Higher ratios and lower grades indicate sediment conditions that are likely unfavorable for invertebrates. The station average combines the PEL ratios across all contaminants measured at a station.

## Value

A [leaflet](#) object

## Examples

```
show_sedimentpelmap(sedimentdata)
```

---

show_segmatrix	<i>Create a colored table for water quality outcomes and exceedances by segment</i>
----------------	---

---

### Description

Create a colored table for water quality outcomes by segment that includes the management action and chlorophyll, and light attenuation exceedances

### Usage

```
show_segmatrix(
  epcdata,
  txtsz = 3,
  trgs = NULL,
  yrrng = c(1975, 2023),
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  abbrev = FALSE,
  family = NA,
  historic = TRUE,
  plotly = FALSE,
  partialyr = FALSE,
  width = NULL,
  height = NULL
)
```

### Arguments

epcdata	data frame of epc data returned by <a href="#">read_importtwq</a>
txtsz	numeric for size of text in the plot, applies only if tab = FALSE
trgs	optional data.frame for annual bay segment water quality targets, defaults to <a href="#">targets</a>
yrrng	numeric vector indicating min, max years to include
bay_segment	chr string for bay segments to include, only one of "OTB", "HB", "MTB", "LTB"
abbrev	logical indicating if text labels in the plot are abbreviated as the first letter
family	optional chr string indicating font family for text labels
historic	logical if historic data are used from 2005 and earlier
plotly	logical if matrix is created using plotly
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

**Details**

This function provides a combined output for the [show\\_wqmatrix](#) and [show\\_matrix](#) functions. Only one bay segment can be plotted for each function call.

**Value**

A static [ggplot](#) object is returned

**See Also**

[show\\_wqmatrix](#), [show\\_matrix](#)

**Examples**

```
show_segmatrix(epcdata, bay_segment = 'OTB')
```

---

show\_segplotly

*Plot chlorophyll and secchi data together with matrix outcomes*

---

**Description**

Plot chlorophyll and secchi data together with matrix outcomes

**Usage**

```
show_segplotly(
  epcdata,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  yrrng = c(1975, 2023),
  family = NULL,
  partialyr = FALSE,
  width = NULL,
  height = NULL
)
```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importwq</a>
bay_segment	chr string for the bay segment, one of "OTB", "HB", "MTB", "LTB"
yrrng	numeric for year range to plot
family	optional chr string indicating font family for text labels
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
width	numeric for width of the plot in pixels
height	numeric for height of the plot in pixels

**Details**

This function combines outputs from [show\\_thrplot](#) and [show\\_segmatrix](#) for a selected bay segment. The plot is interactive and can be zoomed by dragging the mouse pointer over a section of the plot. Information about each cell or value can be seen by hovering over a location in the plot.

**Value**

An interactive plotly object

**Examples**

```
show_segplotly(epcdata)
```

---

```
show_sitemap
```

*Map site attainment categories for a selected year*

---

**Description**

Map site attainment categories for a selected year

**Usage**

```
show_sitemap(
  epcdata,
  yrsel,
  mosel = c(1, 12),
  param = c("chla", "la"),
  trgs = NULL,
  thrs = FALSE,
  partialyr = FALSE
)
```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importtwq</a>
yrsel	numeric for year to plot
mosel	optional numeric of length one or two for mapping results for a specific month or month range in a given year, default full year
param	chr string for which parameter to plot, one of "chla" for chlorophyll or "la" for light attenuation
trgs	optional data.frame for annual bay segment water quality targets, defaults to <a href="#">targets</a> , only applies if mosel = c(1, 12)
thrs	logical indicating if attainment category is relative to targets (default) or thresholds, passed to <a href="#">anlz_attainsite</a> , only applies if mosel = c(1, 12)
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter, only applies if mosel = c(1, 12)

**Value**

A static ggplot object is returned

**Examples**

```
show_sitemap(epcdata, yrsel = 2023)
```

---

show_siteseqmap	<i>Map site and bay segment attainment categories for a selected year</i>
-----------------	---

---

**Description**

Map site and bay segment attainment categories for a selected year

**Usage**

```
show_siteseqmap(
  epcdata,
  yrsel,
  param = c("chla", "la"),
  trgs = NULL,
  thrs = FALSE,
  partialyr = FALSE,
  base_size = 12,
  family = NA
)
```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importwq</a>
yrsel	numeric for year to plot
param	chr string for which parameter to plot, one of "chla" for chlorophyll or "la" for light attenuation
trgs	optional data.frame for annual bay segment water quality targets, defaults to <a href="#">targets</a>
thrs	logical indicating if attainment category is relative to targets (default) or thresholds, passed to <a href="#">anlz_attainsite</a>
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
base_size	numeric indicating text scaling size for plot
family	optional chr string indicating font family for text labels

**Details**

The map is similar to that returned by [show\\_sitemap](#) with the addition of polygons for each bay segment colored by the annual attainment category and the site points are sized relative to the selected parameter in param.

**Value**

A static ggplot object is returned

**Examples**

```
show_sitesegmap(epcdata, yrse1 = 2023)
```

---

show_tbbimatrix	<i>Plot a matrix of Tampa Bay Benthic Index scores over time by bay segment</i>
-----------------	---

---

**Description**

Plot a matrix of Tampa Bay Benthic Index scores over time by bay segment

**Usage**

```
show_tbbimatrix(  
  tbbisr,  
  bay_segment = c("HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"),  
  yrrng = c(1993, 2022),  
  alph = 1,  
  txtsz = 3,  
  family = NA,  
  rev = FALSE,  
  position = "top",  
  plotly = FALSE,  
  width = NULL,  
  height = NULL  
)
```

**Arguments**

tbbisr	input data frame as returned by <a href="#">anlz_tbbisr</a>
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB", "All", "All (wt)"
yrrng	numeric indicating year ranges to evaluate
alph	numeric indicating alpha value for score category colors
txtsz	numeric for size of text in the plot
family	optional chr string indicating font family for text labels

rev	logical if factor levels for bay segments are reversed
position	chr string of location for bay segment labels, default on top, passed to <a href="#">scale_x_discrete</a>
plotly	logical if matrix is created using plotly
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

### Details

Additional summaries are provided for the entire bay, as a summary across categories ("All") and a summary weighted across the relative sizes of each bay segment ("All (wt)").

### Value

A [ggplot](#) object showing trends over time in TBBi scores for each bay segment if plotly = FALSE, otherwise a [plotly](#) object

### Examples

```
tbbiscr <- anlz_tbbiscr(benthicdata)
show_tbbimatrix(tbbiscr)
```

---

show_tbnimatrix	<i>Plot a matrix of Tampa Bay Nekton Index scores over time by bay segment</i>
-----------------	--

---

### Description

Plot a matrix of Tampa Bay Nekton Index scores over time by bay segment

### Usage

```
show_tbnimatrix(
  tbniscr,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  perc = c(32, 46),
  alph = 1,
  txtsz = 3,
  family = NA,
  rev = FALSE,
  position = "top",
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

**Arguments**

tbniscr	input data frame as returned by <a href="#">anzl_tbniscr</a>
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB"
perc	numeric values indicating break points for score categories
alph	numeric indicating alpha value for score category colors
txtsz	numeric for size of text in the plot
family	optional chr string indicating font family for text labels
rev	logical if factor levels for bay segments are reversed
position	chr string of location for bay segment labels, default on top, passed to <a href="#">scale_x_discrete</a>
plotly	logical if matrix is created using plotly
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

**Value**

A [ggplot](#) object showing trends over time in TBNI scores for each bay segment

**Examples**

```
tbniscr <- anzl_tbniscr(fimdata)
show_tbnimatrix(tbniscr)
```

---

show\_tbniscr

*Plot Tampa Bay Nekton Index scores over time by bay segment*

---

**Description**

Plot Tampa Bay Nekton Index scores over time by bay segment

**Usage**

```
show_tbniscr(
  tbniscr,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  perc = c(32, 46),
  alph = 1,
  ylim = c(22, 58),
  rev = FALSE,
  plotly = FALSE,
  family = NA,
  width = NULL,
  height = NULL
)
```



**Arguments**

tbniscr	input dat frame as returned by <a href="#">anlz_tbniscr</a>
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB"
perc	numeric values indicating break points for score categories
alph	numeric indicating alpha value for score category colors
ylim	numeric for y axis limits
rev	logical if factor levels for bay segments are reversed
plotly	logical if matrix is created using plotly
family	optional chr string indicating font family for text labels
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

**Value**

A [ggplot](#) object showing trends over time in TBNI scores for each bay segment or a [plotly](#) object if plotly = TRUE

**Examples**

```
tbniscr <- anlz_tbniscr(fimdata)
show_tbniscr(tbniscr)
```

---

show_tbniscrall	<i>Plot Tampa Bay Nekton Index scores over time as average across bay segments</i>
-----------------	--

---

**Description**

Plot Tampa Bay Nekton Index scores over time as average across bay segments

**Usage**

```
show_tbniscrall(
  tbniscr,
  perc = c(32, 46),
  alph = 1,
  ylim = c(22, 58),
  rev = FALSE,
  plotly = FALSE
)
```

**Arguments**

tbniscr	input dat frame as returned by <a href="#">anlz_tbniscr</a>
perc	numeric values indicating break points for score categories
alph	numeric indicating alpha value for score category colors
ylim	numeric for y axis limits
rev	logical if factor levels for bay segments are reversed
plotly	logical if matrix is created using plotly

**Value**

A [ggplot](#) object showing trends over time in TBNI scores for each bay segment or a [plotly](#) object if `plotly = TRUE`

**Examples**

```
tbniscr <- anlz_tbniscr(fimdata)
show_tbniscrall(tbniscr)
```

---

`show_tbniscrplotly`      *Creates a plotly object for TBNI score plots*

---

**Description**

Creates a plotly object for TBNI score plots

**Usage**

```
show_tbniscrplotly(p, width = NULL, height = NULL)
```

**Arguments**

p	<a href="#">ggplot</a> object as output from <a href="#">show_tbniscr</a> or <a href="#">show_tbniscrall</a>
width	numeric for width of the plot in pixels
height	numeric for height of the plot in pixels

**Value**

A [plotly](#) data object

**Examples**

```
tbniscr <- anlz_tbniscr(fimdata)
p <- show_tbniscrall(tbniscr)
show_tbniscrplotly(p)
```

---

show_tdlcrk	<i>Make a map for tidal creek report card</i>
-------------	---

---

**Description**

Make a map for tidal creek report card

**Usage**

```
show_tdlcrk(dat, weight = 1.5)
```

**Arguments**

dat	input creek score data returned from <a href="#">anz_tdlcrk</a>
weight	numeric for weight of polylines, passed to <a href="#">addPolylines</a>

**Value**

A [leaflet](#) object

**Examples**

```
dat <- anz_tdlcrk(tidalcreeks, iwrraw, yr = 2023)
show_tdlcrk(dat)
```

---

show_tdlcrkindic	<i>Plotly barplots of tidal creek context indicators</i>
------------------	--

---

**Description**

Plotly barplots of tidal creek context indicators

**Usage**

```
show_tdlcrkindic(
  id,
  cntdat,
  yr = 2023,
  thrse1 = FALSE,
  pal = c("#5C4A42", "#427355", "#004F7E")
)
```

**Arguments**

id	numeric indicating the id number of the tidal creek to plot
cntdat	output from <a href="#">anlz_tdlcrkindic</a>
yr	numeric indicating reference year
thrse1	logical if threshold lines and annotations are shown on the plots
pal	vector of colors for the palette

**Value**

A plotly object

**Examples**

```
cntdat <- anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2023)

set.seed(123)
id <- sample(unique(cntdat$id), 1)
show_tdlcrkindic(id, cntdat, thrse1 = TRUE)
```

---

show\_tdlcrkindiccdf *Plotly empirical CDF plots of tidal creek context indicators*

---

**Description**

Plotly empirical CDF plots of tidal creek context indicators

**Usage**

```
show_tdlcrkindiccdf(
  id,
  cntdat,
  yr = 2023,
  thrse1 = FALSE,
  pal = c("#5C4A42", "#427355", "#004F7E")
)
```

**Arguments**

id	numeric indicating the id number of the tidal creek to plot
cntdat	output from <a href="#">anlz_tdlcrkindic</a>
yr	numeric indicating reference year
thrse1	logical if threshold lines and annotations are shown on the plots
pal	vector of colors for the palette

**Details**

This function returns several empirical cumulative distribution plots for the tidal creek context indicators. Points on the plot indicate the observed values and percentiles for the creek specified by `id`. The percentiles and CDF values are defined by the "population" of creeks in `cntdat`. Points in the plots are color-coded by sample year to evaluate temporal trends, if any.

**Value**

A plotly object

**Examples**

```
cntdat <- anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2023)

set.seed(123)
id <- sample(unique(cntdat$id), 1)
show_tdlcrkindiccdf(id, cntdat, thrsel = TRUE)
```

---

show_tdlcrkline	<i>Add a line or annotation to a plotly graph</i>
-----------------	---

---

**Description**

Add a line or annotation to a plotly graph for the tidal creek indicators

**Usage**

```
show_tdlcrkline(
  varin = c("CHLAC", "TN", "chla_tn_ratio", "DO", "tsi", "no23_ratio"),
  thrsel = FALSE,
  horiz = TRUE,
  annotate = FALSE
)
```

**Arguments**

<code>varin</code>	chr string for the indicator
<code>thrsel</code>	logical if something is returned, otherwise NULL, this is a hack for working with the plotly output
<code>horiz</code>	logical indicating if output is horizontal or vertical
<code>annotate</code>	logical indicating if output is line or annotation text

**Details**

This function is used internally within [show\\_tdlcrkindic](#) and [show\\_tdlcrkindiccdf](#)

**Value**

A list object passed to the layout argument of plotly, either shapes or annotate depending on user input

**Examples**

```
# code for vertical line output, chloropyll
show_tdlcrkline('CHLAC', thrsel = TRUE)
```

---

show_tdlcrkmatrix	<i>Plot the tidal creek report card matrix</i>
-------------------	--

---

**Description**

Plot the tidal creek report card matrix

**Usage**

```
show_tdlcrkmatrix(
  dat,
  class = c("3M", "2"),
  score = c("Prioritize", "Investigate", "Caution", "Monitor"),
  family = NA,
  size = 11
)
```

**Arguments**

dat	input creek score data returned from <a href="#">anz_tdlcrk</a>
class	character vector indicating which creek classes to show, one to many of '3M', '2', '3F', and '1'. Defaults to marine only ('3M', '2').
score	character vector of score categories to include, one to many of 'Prioritize', 'Investigate', 'Caution', and 'Monitor'. Defaults to all.
family	optional chr string indicating font family for text labels
size	numeric for text and line scaling

**Details**

The plot shows a matrix with rows for individual creeks and columns for overall creek score. The columns show an overall creek score and the number of years in the prior ten years that nitrogen values at a creek were assigned to each of the four score categories. Number of years is mapped to cell transparency.

**Value**

A static [ggplot](#) object is returned.

## Examples

```
dat <- anlz_tdlcrk(tidalcreeks, iwrraw, yr = 2023)
show_tdlcrkmatrix(dat)
```

---

show\_tdlcrkradar      *Radar plots for tidal creek indicators*

---

## Description

Radar plots for tidal creek indicators

## Usage

```
show_tdlcrkradar(
  id,
  cntdat,
  col = "#338080E6",
  ptsz = 1,
  lbsz = 0.8,
  valsz = 1,
  brdwd = 5
)
```

## Arguments

id	numeric indicating the id number of the tidal creek to plot
cntdat	output from <a href="#">anlz_tdlcrkindic</a>
col	color input for polygon and line portions
ptsz	numeric size of points
lbsz	numeric for size of text labels
vals	numeric for size of numeric value labels
brdwd	numeric for polygon border width

## Details

See details in [anlz\\_tdlcrkindic](#) for an explanation of the indicators

Internal code borrowed heavily from the radarchart function in the fmsb package.

## Value

A radar plot

**Examples**

```

cntdat <- anlz_tdlcrkindic(tidalcreeks, iwrraw, yr = 2023, radar = TRUE)

set.seed(123)
id <- sample(unique(cntdat$id), 1)
show_tdlcrkradar(id, cntdat)

```

---

show_thrplot	<i>Plot annual water quality values, targets, and thresholds for a segment</i>
--------------	--

---

**Description**

Plot annual water quality values, targets, and thresholds for a bay segment

**Usage**

```

show_thrplot(
  epcdata,
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  thr = c("chl", "la"),
  trgs = NULL,
  yrrng = c(1975, 2023),
  family = NA,
  labelexp = TRUE,
  txtlab = TRUE,
  thrs = FALSE,
  partialyr = FALSE
)

```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importwq</a>
bay_segment	chr string for the bay segment, one of "OTB", "HB", "MTB", "LTB"
thr	chr string indicating which water quality value and appropriate target/threshold to plot, one of "chl" for chlorophyll and "la" for light availability
trgs	optional data.frame for annual bay segment water quality targets/thresholds, defaults to <a href="#">targets</a>
yrrng	numeric vector indicating min, max years to include
family	optional chr string indicating font family for text labels
labelexp	logical indicating if y axis and target labels are plotted as expressions, default TRUE
txtlab	logical indicating if a text label for the target value is shown in the plot
thrs	logical indicating if reference lines are shown only for the regulatory threshold
partialyr	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter



**Value**

A [ggplot](#) object

**Examples**

```
show_thrplot(epcdata, bay_segment = 'OTB', thr = 'chl')
```

---

```
show_transect
```

*Plot results for a seagrass transect by time and location*

---

**Description**

Plot results for a seagrass transect by time and location

**Usage**

```
show_transect(
  transect,
  site,
  species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia", "Caulerpa",
    "Dapis"),
  yrrng = c(1998, 2023),
  varplo = c("Abundance", "Blade Length", "Short Shoot Density"),
  base_size = 12,
  facet = FALSE,
  ncol = NULL,
  plotly = FALSE,
  width = NULL,
  height = NULL,
  sppcol = NULL
)
```

**Arguments**

transect	data frame returned by <a href="#">read_transect</a>
site	chr string indicating site results to plot
species	chr string indicating one to many of which species to plot
yrrng	numeric indicating year ranges to evaluate
varplo	chr string indicating which variable to plot
base_size	numeric indicating text scaling size for plot
facet	logical indicating if plots are separated into facets by species
ncol	numeric indicating number of columns if facet = TRUE
plotly	logical if plot is created using plotly
width	numeric for width of the plot in pixels, only applies if plotly = TRUE

height	numeric for height of the plot in pixels, only applies if plotly = TRUE
sppcol	character vector of alternative colors to use for each species, must have length of six

### Details

All sites along a transect that were surveyed are shown in the plot, including those where the selected species were not found. The latter is colored in grey hollow points. Species options include Halodule, Syringodium, Thalassia, Halophila, Ruppia, Caulerpa (attached macroalgae), and/or Dapis (cyanobacteria). Drift or attached macroalgae and cyanobacteria (Dapis) estimates may not be accurate prior to 2021.

Note that if plotly = TRUE, the size legend is not shown.

### Value

A `ggplot` object

### Examples

```
## Not run:
transect <- read_transect()

## End(Not run)

# one species
show_transect(transect, site = 'S3T10', species = 'Halodule', varplo = 'Abundance')

# multiple species, one plot
show_transect(transect, site = 'S3T10',
  species = c('Halodule', 'Syringodium', 'Thalassia', 'Halophila', 'Ruppia',
    'Caulerpa', 'Dapis'),
  varplo = 'Abundance')

# multiple species, multiple plots
show_transect(transect, site = 'S3T10',
  species = c('Halodule', 'Syringodium', 'Thalassia', 'Halophila', 'Ruppia',
    'Caulerpa', 'Dapis'),
  varplo = 'Abundance', facet = TRUE)
```

---

show_transectavespp	<i>Show annual averages of seagrass frequency occurrence by bay segments, year, and species</i>
---------------------	---

---

### Description

Show annual averages of seagrass frequency occurrence by bay segments, year, and species

**Usage**

```
show_transectavespp(
  transectocc,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCB"),
  yrrng = c(1998, 2023),
  species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia", "Caulerpa",
    "Dapis"),
  total = TRUE,
  alph = 1,
  family = NA,
  plotly = FALSE,
  asreact = FALSE,
  width = NULL,
  height = NULL,
  sppcol = NULL
)
```

**Arguments**

transectocc	data frame returned by <a href="#">anzl_transectocc</a>
bay_segment	chr string for the bay segment, one to many of "OTB", "HB", "MTB", "LTB", "BCB"
yrrng	numeric indicating year ranges to evaluate
species	chr string of species to summarize, one to many of "Halodule", "Syringodium", "Thalassia", "Ruppia", "Halophila", "Caulerpa", "Dapis"
total	logical indicating if total frequency occurrence for all species is also returned, only applies if asreact = FALSE
alph	numeric indicating alpha value for score category colors
family	optional chr string indicating font family for text labels
plotly	logical if matrix is created using plotly
asreact	logical if a reactable table is returned instead of a plot
width	numeric for width of the plot in pixels, only applies if plotly = TRUE
height	numeric for height of the plot in pixels, only applies if plotly = TRUE
sppcol	character vector of alternative colors to use for each species, must have length of six

**Details**

Results are based on averages across species by date and transect in each bay segment. Drift or attached macroalgae (e.g., *Caulerpa*) and cyanobacteria (*Dapis*) estimates may not be accurate prior to 2021.

**Value**

If asreact = F, a [ggplot](#) or [plotly](#) (if plotly = T) object is returned showing trends over time by species for selected bay segments. If asreact = T, a [reactable](#) table showing results by year, segment, and species is returned.

## References

The plot is a representation of figure 2 in Johansson, R. (2016) Seagrass Transect Monitoring in Tampa Bay: A Summary of Findings from 1997 through 2015, Technical report #08-16, Tampa Bay Estuary Program, St. Petersburg, Florida.

The table is a representation of table 2, p. 163 in Yarbrow, L. A., and P. R. Carlson, Jr., eds. 2016. Seagrass Integrated Mapping and Monitoring Program: Mapping and Monitoring Report No. 2. Fish and Wildlife Research Institute Technical Report TR-17 version 2. vi + 281 p.

## Examples

```
## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
show_transectavespp(transectocc)
```

---

show_transectmatrix	<i>Show matrix of seagrass frequency occurrence by bay segments and year</i>
---------------------	--

---

## Description

Show matrix of seagrass frequency occurrence by bay segments and year

## Usage

```
show_transectmatrix(
  transectocc,
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCB"),
  total = TRUE,
  neutral = FALSE,
  yrrng = c(1998, 2023),
  alph = 1,
  txtsz = 3,
  family = NA,
  rev = FALSE,
  position = "top",
  plotly = FALSE,
  width = NULL,
  height = NULL
)
```

**Arguments**

transectocc	data frame returned by <a href="#">anlz_transectocc</a>
bay_segment	chr string for the bay segment, one to many of "HB", "OTB", "MTB", "LTB", "TCB", "MR", "BCB"
total	logical indicating if average frequency occurrence is calculated for the entire bay across segments
neutral	logical indicating if a neutral and continuous color scheme is used
yrrng	numeric indicating year ranges to evaluate
alph	numeric indicating alpha value for score category colors
txtsz	numeric for size of text in the plot
family	optional chr string indicating font family for text labels
rev	logical if factor levels for bay segments are reversed
position	chr string of location for bay segment labels, default on top, passed to <a href="#">scale_x_discrete</a>
plotly	logical if matrix is created using plotly
width	numeric for width of the plot in pixels, only applies of plotly = TRUE
height	numeric for height of the plot in pixels, only applies of plotly = TRUE

**Details**

Results are based on averages across species by date and transect in each bay segment

The color scheme is based on arbitrary breaks at 25, 50, and 75 percent frequency occurrence. These don't necessarily translate to any ecological breakpoints. Use `neutral = TRUE` to use a neutral and continuous color palette.

**Value**

A [ggplot](#) object showing trends over time for each bay segment if `plotly = FALSE`, otherwise a [plotly](#) object

**References**

This plot is a representation of Table 1 in R. Johansson (2016) Seagrass Transect Monitoring in Tampa Bay: A Summary of Findings from 1997 through 2015, Technical report #08-16, Tampa Bay Estuary Program, St. Petersburg, Florida.

**Examples**

```
## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
show_transectmatrix(transectocc)
```

---

show_transectsum	<i>Plot frequency occurrence for a seagrass transect by time for all species</i>
------------------	--

---

## Description

Plot frequency occurrence for a seagrass transect by time for all species

## Usage

```
show_transectsum(
  transectocc,
  site,
  species = c("Halodule", "Syringodium", "Thalassia", "Halophila", "Ruppia", "Caulerpa",
    "Dapis"),
  yrrng = c(1998, 2023),
  abund = FALSE,
  sppcol = NULL
)
```

## Arguments

transectocc	data frame returned by <a href="#">anz_transectocc</a>
site	chr string indicating site results to plot
species	chr string indicating which species to plot
yrrng	numeric indicating year ranges to evaluate
abund	logical indicating if abundance averages are plotted instead of frequency occurrence
sppcol	character vector of alternative colors to use for each species, must have length of six

## Details

This plot provides a quick visual assessment of how frequency occurrence or abundance for multiple species has changed over time at a selected transect. Drift or attached macroalgae (e.g., *Caulerpa*) and cyanobacteria (*Dapis*) estimates may not be accurate prior to 2021.

## Value

A [plotly](#) object

**Examples**

```
## Not run:
transect <- read_transect()

## End(Not run)
transectocc <- anlz_transectocc(transect)
show_transectsum(transectocc, site = 'S3T10')
```

---

show_wqmatrix	<i>Create a colored table for chlorophyll or light attenuation exceedances</i>
---------------	--

---

**Description**

Create a colored table for chlorophyll or light attenuation exceedances

**Usage**

```
show_wqmatrix(
  epcdata,
  param = c("chla", "la"),
  txtsz = 3,
  trgs = NULL,
  yrrng = c(1975, 2023),
  bay_segment = c("OTB", "HB", "MTB", "LTB"),
  asreact = FALSE,
  nrows = 10,
  abbrev = FALSE,
  family = NA,
  plotly = FALSE,
  partialyr = FALSE,
  width = NULL,
  height = NULL
)
```

**Arguments**

epcdata	data frame of epc data returned by <a href="#">read_importwq</a>
param	chr string for which parameter to plot, one of "chla" for chlorophyll or "la" for light attenuation
txtsz	numeric for size of text in the plot, applies only if tab = FALSE
trgs	optional data.frame for annual bay segment water quality targets, defaults to <a href="#">targets</a>
yrrng	numeric vector indicating min, max years to include
bay_segment	chr string for bay segments to include, one to all of "OTB", "HB", "MTB", "LTB"

<code>asreact</code>	logical indicating if a <a href="#">reactable</a> object is returned
<code>nrows</code>	if <code>asreact = TRUE</code> , a numeric specifying number of rows in the table
<code>abbrev</code>	logical indicating if text labels in the plot are abbreviated as the first letter
<code>family</code>	optional chr string indicating font family for text labels
<code>plotly</code>	logical if matrix is created using plotly
<code>partialyr</code>	logical indicating if incomplete annual data for the most recent year are approximated by five year monthly averages for each parameter
<code>width</code>	numeric for width of the plot in pixels, only applies of <code>plotly = TRUE</code>
<code>height</code>	numeric for height of the plot in pixels, only applies of <code>plotly = TRUE</code>

**Value**

A static [ggplot](#) object is returned if `asreact = FALSE`, otherwise a [reactable](#) table is returned

**See Also**

[show\\_matrix](#), [show\\_segmatrix](#)

**Examples**

```
show_wqmatrix(epcdata)
```

---

stations	<i>Bay stations by segment</i>
----------	--------------------------------

---

**Description**

Bay stations by segment

**Usage**

```
stations
```

**Format**

A data frame with 45 rows and 4 variables:

**bay\_segment** chr  
**epchc\_station** num  
**Latitude** num  
**Longitude** num



---

subtacres	<i>Tampa Bay subtidal cover</i>
-----------	---------------------------------

---

**Description**

Tampa Bay subtidal cover for Habitat Master Plan reporting

**Usage**

```
subtacres
```

**Format**

A data frame

**Examples**

```
## Not run:  
load(url("https://github.com/tbep-tech/hmpu-workflow/raw/master/data/subtacres.RData"))  
  
save(subtacres, file = 'data/subtacres.RData', compress = 'xz')  
  
## End(Not run)
```

---

swfwmdtbseg	<i>Spatial data object of SWFWMD Tampa Bay segments</i>
-------------	---

---

**Description**

Spatial data object of SWFWMD Tampa Bay segments

**Usage**

```
swfwmdtbseg
```

**Format**

A simple features `sf` object (MULTIPOLYGON) with 7 features and 1 fields, `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`

**Examples**

```
## Not run:
library(sf)
library(dplyr)

levs <- c('oldTampaBay', 'hillsboroughBay', 'middleTampaBay',
          'lowerTampaBay', 'bocaCiegaBay', 'terraCieaBay', 'manateeRiver')
labs <- c('Old Tampa Bay', 'Hillsborough Bay', 'Middle Tampa Bay',
          'Lower Tampa Bay', 'Boca Ciega Bay', 'Terra Ceia Bay', 'Manatee River')

swfwmdtbseg <- st_read(
  dsn = 'T:/05_GIS/SWFWM/Seagrass/2022_Seagrass/DraftMaps2022_1130.gdb',
  layer = 'suncoastSeagrassSegments'
) %>%
  filter(waterbodyName %in% levs) %>%
  mutate(
    waterbodyName = factor(waterbodyName, levels = levs, labels = labs)
  ) %>%
  select(segment = waterbodyName) %>%
  st_transform(crs = 4326)

save(swfwmdtbseg, file = 'data/swfwmdtbseg.RData', compress = 'xz')

## End(Not run)
```

---

 targets

*Bay segment targets*


---

**Description**

Bay segment specific management targets including low and high magnitude exceedance thresholds

**Usage**

```
targets
```

**Format**

A data frame with 4 rows and 8 variables:

```
bay_segment chr
name chr
chla_target num
chla_smallex num
chla_thresh num
la_target num
la_smallex num
la_thresh num
```

**Examples**

```
## Not run:

targets <- structure(list(
  bay_segment = c("OTB", "HB", "MTB", "LTB", "BCBN", "BCBS", "TCB", "MR", "RALTB"),
  name = c("Old Tampa Bay", "Hillsborough Bay", "Middle Tampa Bay", "Lower Tampa Bay",
    "Boca Ciega Bay North", "Boca Ciega Bay South", "Terra Ceia Bay", "Manatee River",
    "Remainder Lower Tampa Bay"),
  chla_target = c(8.5, 13.2, 7.4, 4.6, 7.7, 6.1, 7.5, 7.3, NaN),
  chla_smalllex = c(8.9, 14.1, 7.9, 4.8, NaN, NaN, NaN, NaN, NaN),
  chla_thresh = c(9.3, 15, 8.5, 5.1, 8.3, 6.3, 8.7, 8.8, 5.1),
  la_target = c(0.83, 1.58, 0.83, 0.63, NaN, NaN, NaN, NaN, NaN),
  la_smalllex = c(0.86, 1.63, 0.87, 0.66, NaN, NaN, NaN, NaN, NaN),
  la_thresh = c(0.88, 1.67, 0.91, 0.68, NaN, NaN, NaN, NaN, NaN)),
  class = "data.frame", row.names = c(NA, -9L)
)

save(targets, file = 'data/targets.RData')

## End(Not run)
```

tbniref

*Reference conditions for Tampa Bay Nekton Index metrics***Description**

Reference conditions for Tampa Bay Nekton Index metrics

**Usage**

```
tbniref
```

**Format**

A data frame with 16 rows and 12 variables:

```
bay_segment chr
Season chr
NumTaxa_P5 num
NumTaxa_P95 num
BenthicTaxa_P5 num
BenthicTaxa_P95 num
TaxaSelect_P5 num
TaxaSelect_P95 num
NumGuilds_P5 num
NumGuilds_P95 num
Shannon_P5 num
Shannon_P95 num
```

**Examples**

```
## Not run:

library(tbeptools)

tbniref <- anlz_tbnimet(fimdata) %>%
  dplyr::filter(between(Year, 1998, 2015)) %>%
  dplyr::select(Season, bay_segment, NumTaxa, BenthicTaxa, TaxaSelect, NumGuilds, Shannon) %>%
  dplyr::group_by(bay_segment, Season) %>%
  dplyr::summarize(NumTaxa_P5 = round(quantile(NumTaxa, probs = 0.05)),
                  NumTaxa_P95 = round(quantile(NumTaxa, probs = 0.95)),
                  BenthicTaxa_P5 = round(quantile(BenthicTaxa, probs = 0.05)),
                  BenthicTaxa_P95 = round(quantile(BenthicTaxa, probs = 0.95)),
                  TaxaSelect_P5 = round(quantile(TaxaSelect, probs = 0.05)),
                  TaxaSelect_P95 = round(quantile(TaxaSelect, probs = 0.95)),
                  NumGuilds_P5 = round(quantile(NumGuilds, probs = 0.05)),
                  NumGuilds_P95 = round(quantile(NumGuilds, probs = 0.95)),
                  Shannon_P5 = quantile(Shannon, probs = 0.05),
                  Shannon_P95 = quantile(Shannon, probs = 0.95))

save(tbniref, file = 'data/tbniref.RData', compress = 'xz')

## End(Not run)
```

---

tbnispp

*Reference table for Tampa Bay Nekton Index species classifications*


---

**Description**

Reference table for Tampa Bay Nekton Index species classifications

**Usage**

```
tbnispp
```

**Format**

A data frame with 196 rows and 10 variables:

**TSN** int

**NODCCODE** chr

**ScientificName** chr

**Include\_TB\_Index** chr

**Hab\_Cat** chr

**Est\_Cat** chr

**Est\_Use** chr

**Feeding\_Cat** chr  
**Feeding\_Guild** chr  
**Selected\_Taxa** chr

### Examples

```
## Not run:
library(dplyr)

# import and clean
tbnispp <- read.csv('../tbni-proc/data/TBIndex_spp_codes.csv',
  header = TRUE, stringsAsFactors = FALSE) %>%
  mutate(
    NODCCODE = as.character(NODCCODE),
    NODCCODE = case_when(NODCCODE == "9.998e+09" ~ "9998000000",
      TRUE ~ NODCCODE)
  )

save(tbnispp, file = 'data/tbnispp.RData', compress = 'xz')

## End(Not run)
```

---

tbseg

*Spatial data object of Tampa Bay segments*

---

### Description

Note that these boundaries are not used for formal analysis and are only used as visual aids in mapping.

### Usage

```
tbseg
```

### Format

A simple features `sf` object (POLYGON) with 4 features and 2 fields, `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`

**long\_name** chr  
**bay\_segment** chr

### Details

Spatial data object of Tampa Bay segments

### Examples

```
library(sf)
plot(st_geometry(tbseg))
```

---

 tbsegdetail

*Spatial data object of detailed Tampa Bay segments*


---

### Description

Note that these boundaries are not used for formal analysis and are only used as visual aids in mapping. The data object differs from `tbseg` by including Boca Ciega Bay, Terra Ceia Bay, and Manatee River. The boundaries are also more detailed.

### Usage

```
tbsegdetail
```

### Format

A simple features `sf` object (POLYGON) with 7 features and 2 fields, `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`

**long\_name** chr

**bay\_segment** chr

### Details

Spatial data object of detailed Tampa Bay segments

### Examples

```
library(sf)
plot(st_geometry(tbsegdetail))
```

---

 tbseglines

*Spatial data object of lines defining major Tampa Bay segments*


---

### Description

Spatial data object of lines defining major Tampa Bay segments

### Usage

```
tbseglines
```

### Format

A simple features `sf` object (LINESTRING) with 3 features and 1 field, `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`

**Examples**

```
library(sf)
plot(st_geometry(tbseglines))
```

---

tbsegshed	<i>Spatial data object of Tampa Bay segments plus watersheds</i>
-----------	--

---

**Description**

Spatial data object of Tampa Bay segments plus watersheds

**Usage**

```
tbsegshed
```

**Format**

A simple features [sf](#) object (POLYGON) with 7 features and 2 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs

**long\_name** chr

**bay\_segment** chr

**Examples**

```
library(sf)
plot(st_geometry(tbsegshed))
```

---

tbshed	<i>Spatial data object of Tampa Bay watershed</i>
--------	---

---

**Description**

Spatial data object of Tampa Bay watershed, includes the bay proper

**Usage**

```
tbshed
```

**Format**

A simple features [sf](#) object (POLYGON) with 1 feature and 0 fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs

## Examples

```
library(sf)
## Not run:
library(dplyr)
tbshed <- st_read('T:/05_GIS/BOUNDARIES/TBEP_Watershed_Correct_Projection.shp') %>%
  select(-Id, -Name, -Area_m, -Hectares) %>%
  st_transform(crs = 4326)

save(tbshed, file = 'data/tbshed.RData', compress = 'xz')

## End(Not run)
plot(st_geometry(tbshed))
```

---

tidalcreeks

*Spatial data object of tidal creeks in Impaired Waters Rule, Run 65*

---

## Description

Spatial data object of tidal creeks in Impaired Waters Rule, Run 65

## Usage

```
tidalcreeks
```

## Format

A simple features `sf` object (MULTILINESTRING) with 615 features and fields, +proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs

**id** num

**wbid** chr

**JEI** chr

**class** chr

**name** chr

**Creek\_Length\_m** num

## Details

File was created using workflow at [https://tbep-tech.github.io/tidalcreek-stats/Creek\\_select\\_tbeptools](https://tbep-tech.github.io/tidalcreek-stats/Creek_select_tbeptools), example below is old and for Run 61.



**Examples**

```
## Not run:
library(sf)
library(dplyr)

prj <- 4326

# create sf object of creek population, join with creek length data
tidalcreeks <- st_read(
  dsn = '../..02_DOCUMENTS/tidal_creeks/TidalCreek_ALL_Line_WBID61.shp',
  drivers = 'ESRI Shapefile'
) %>%
  st_transform(crs = prj) %>%
  mutate(
    id = 1:nrow(.)
  ) %>%
  select(id, name = Name, JEI = CreekID, wbid = WBID, class = CLASS, Creek_Length_m = Total_m)

# save
save(tidalcreeks, file = 'data/tidalcreeks.RData', compress = 'xz')

## End(Not run)
```

---

tidaltargets	<i>Tidal creek nitrogen targets</i>
--------------	-------------------------------------

---

**Description**

Tidal creek nitrogen targets

**Usage**

```
tidaltargets
```

**Format**

A data frame with 2 rows and 4 variables:

**region** chr

**prioritize** num

**investigate** num

**caution** num

---

transect

*Seagrass transect data for Tampa Bay current as of 04122024*

---

### Description

Seagrass transect data for Tampa Bay current as of 04122024

### Usage

```
transect
```

### Format

A data frame with 152580 rows and 11 variables:

**Crew** chr

**MonitoringAgency** chr

**Date** Date

**Transect** chr

**Site** chr

**Depth** int

**Savspecies** chr

**SeagrassEdge** num

**var** chr

**aveval** num

**sdval** num

### Examples

```
## Not run:
```

```
transect <- read_transect()
```

```
save(transect, file = 'data/transect.RData', compress = 'xz')
```

```
## End(Not run)
```

---

trnlns	<i>Seagrass transect locations</i>
--------	------------------------------------

---

**Description**

Seagrass transect locations

**Usage**

```
trnlns
```

**Format**

A sf LINESTRING object

**Examples**

```
## Not run:
library(sf)
library(dplyr)

trnlns <- st_read('T:/05_GIS/SEAGRASS_TRANSECTS/transect_routes.shp') %>%
  st_transform(crs = 4326) %>%
  dplyr::filter(!as.character(Site) %in% c('S8T1', 'S8T2', 'S8T3', 'S3T2')) %>%
  dplyr::mutate_if(is.factor, as.character) %>%
  dplyr::filter(Site %in% trnpts$TRAN_ID)

# add bearing, positive counter-clockwise from east
bearing <- lapply(trnlns$geometry, function(x) geosphere::bearing(x[, c(1:2)])) %>%
  unlist()

trnlns$bearing <- bearing

save(trnlns, file = 'data/trnlns.RData', compress = 'xz')

## End(Not run)
```

---

trnpts	<i>Seagrass transect starting locations</i>
--------	---

---

**Description**

Seagrass transect starting locations

**Usage**

```
trnpts
```

**Format**

A sf POINT object

**Examples**

```
## Not run:
library(sf)
library(dplyr)
library(tbeptools)

trnpts <- st_read('T:/05_GIS/SEAGRASS_TRANSECTS/TransectBasics2019.shp') %>%
  st_transform(crs = 4326) %>%
  dplyr::rename(MonAgency = 'MON_AGENCY') %>%
  dplyr::filter(!as.character(TRAN_ID) %in% c('S8T1', 'S8T2', 'S8T3', 'S3T2'))

# s4t10 is slightly outside boundary for lower tampa bay
s4t10 <- trnpts %>%
  dplyr::filter(TRAN_ID %in% 'S4T10') %>%
  dplyr::mutate(bay_segment = 'LTB')

trnpts <- trnpts %>%
  sf::st_intersection(sf::st_make_valid(tbsegshed)) %>%
  dplyr::select(-long_name) %>%
  dplyr::mutate_if(is.factor, as.character) %>%
  dplyr::bind_rows(s4t10)

save(trnpts, file = 'data/trnpts.RData', compress = 'xz')

## End(Not run)
```

---

util\_fibicons

*Return leaflet icon set for FIB maps*

---

**Description**

Return leaflet icon set for FIB maps

**Usage**

```
util_fibicons(indic)
```

**Arguments**

**indic** character indicating "entero", "fcolif", or "fibmat" for *Enterococcus*, Fecal Coliform, or FIB matrix maps, respectively

**Details**

Used internally with [show\\_enteromap](#) for wet/dry icons for *Enterococcus*, with [show\\_fibmap](#) for *E. Coli/Enterococcus* icons (EPCHC data), and with [show\\_fibmatmap](#) for matrix annual score category icons for EPCHC and non-EPCHC data.

**Value**

A leaflet icon set as returned by [iconList](#).

**Examples**

```
util_fibicons(indic = 'entero')
util_fibicons(indic = 'fcolif')
util_fibicons(indic = 'fibmat')
```

---

util_fiblevs	<i>A list of Fecal Indicator Bacteria (FIB) factor levels and labels</i>
--------------	--

---

**Description**

A list of Fecal Indicator Bacteria (FIB) factor levels and labels

**Usage**

```
util_fiblevs()
```

**Value**

A list with levels (often cutpoints) and labels for FIB categories

**Examples**

```
util_fiblevs()
```

---

util_html	<i>Convert character string to html class</i>
-----------	---

---

**Description**

Convert character string to html class

**Usage**

```
util_html(text)
```

**Arguments**

text                    character string input

**Details**

Adapted from [HTML](#)

**Value**

The same input character string with html class

**Examples**

```
util_html('abd')
```

---

util_map	<i>Create an empty leaflet map from sf input</i>
----------	--

---

**Description**

Create an empty leaflet map from sf input

**Usage**

```
util_map(tomap, minimap = "bottomleft")
```

**Arguments**

tomap                    sf input object  
minimap                  character string indicating location of minimap, use minimap = NULL to suppress

**Value**

A leaflet object with optional minimap and ESRI provider tiles

**Examples**

```
tomap <- tibble::tibble(  
  lon = -82.6365,  
  lat = 27.75822  
)  
tomap <- sf::st_as_sf(tomap, coords = c('lon', 'lat'), crs = 4326)  
util_map(tomap)
```

---

util_organ	<i>Get organization name from organization identifier in USEPA Water Quality Portal</i>
------------	---

---

**Description**

Get organization name from organization identifier in USEPA Water Quality Portal

**Usage**

```
util_organ(org, stanm = FALSE)
```

**Arguments**

org	chr string indicating the organization identifier, see <a href="#">read_importwqp</a> for valid entries
stanm	logical indicating if a character string for a column name specific to the organization is returned

**Value**

A character string of the organization name that corresponds to the organization identifier or a column name for the station identifier specific to the organization if stanm = TRUE

**Examples**

```
util_organ('21FLHILL_WQX')  
util_organ('21FLPASC_WQX', stanm = TRUE)
```

# Index

## \* analyze

- anlz\_attain, 5
- anlz\_attainsite, 6
- anlz\_avedat, 7
- anlz\_avedatsite, 7
- anlz\_hmpreport, 14
- anlz\_hydroload, 16
- anlz\_iwrraw, 17
- anlz\_refs, 17
- anlz\_tbbimed, 22
- anlz\_tbbiscr, 23
- anlz\_tbniave, 23
- anlz\_tbnimet, 24
- anlz\_tbniscr, 25
- anlz\_tdlcrk, 26
- anlz\_tdlcrkindic, 26
- anlz\_transectave, 27
- anlz\_transectavespp, 28
- anlz\_transectocc, 30
- anlz\_yrattain, 31

## \* anlz

- anlz\_fibmap, 9
- anlz\_sedimentaddtot, 18
- anlz\_sedimentave, 19
- anlz\_sedimentpel, 20
- anlz\_sedimentpelave, 21

## \* datasets

- acres, 4
- benthicdata, 31
- bsmap, 32
- catchpixels, 32
- catchprecip, 33
- enterodata, 34
- epcdata, 35
- fibdata, 36
- fimdata, 37
- fimstations, 38
- hmptrgs, 39
- iwrraw, 39

- phytodata, 40
- seagrass, 62
- sedimentdata, 63
- sgmanagement, 64
- sgseg, 65
- stations, 112
- subt acres, 113
- swfwmdtbseg, 113
- targets, 114
- tbniref, 115
- tbnispp, 116
- tbseg, 117
- tbsegdetail, 118
- tbseglines, 118
- tbsegshed, 119
- tbshed, 119
- tidalcreeks, 120
- tidaltargets, 121
- transect, 122
- trnlns, 123
- trnpts, 123

## \* data

- acres, 4
- benthicdata, 31
- epcdata, 35
- fibdata, 36
- fimdata, 37
- fimstations, 38
- hmptrgs, 39
- iwrraw, 39
- phytodata, 40
- seagrass, 62
- sedimentdata, 63
- sgmanagement, 64
- sgseg, 65
- stations, 112
- subt acres, 113
- swfwmdtbseg, 113
- tbseg, 117



- tbsegdetail, 118
- tbseglines, 118
- tbsegshed, 119
- tbshed, 119
- tidalcreeks, 120
- transect, 122
- trnlns, 123
- trnpts, 123
- \* read**
  - read\_dlcurrent, 41
  - read\_formbenthic, 42
  - read\_formfib, 43
  - read\_formfim, 44
  - read\_formphyto, 45
  - read\_formsediment, 46
  - read\_formtransect, 47
  - read\_formwq, 48
  - read\_formwqp, 49
  - read\_importbenthic, 50
  - read\_importepc, 52
  - read\_importfib, 53
  - read\_importfim, 55
  - read\_importphyto, 56
  - read\_importsediment, 57
  - read\_importwq, 58
  - read\_importwqp, 60
  - read\_transect, 61
- \* show**
  - anlz\_fibmatrix, 11
  - show\_annualassess, 66
  - show\_boxplot, 67
  - show\_compplot, 68
  - show\_enteromap, 69
  - show\_fibmap, 71
  - show\_fibmatmap, 72
  - show\_fibmatrix, 74
  - show\_hmpreport, 76
  - show\_matrix, 78
  - show\_matrixplotly, 79
  - show\_ratab, 80
  - show\_reactable, 81
  - show\_seagrasscoverage, 82
  - show\_sedimentave, 85
  - show\_sedimentmap, 86
  - show\_sedimentpelave, 87
  - show\_sedimentpelaveplotly, 88
  - show\_sedimentpelmap, 89
  - show\_segmatrix, 90
  - show\_segplotly, 91
  - show\_sitemap, 92
  - show\_sitesegmap, 93
  - show\_tbbimatrix, 94
  - show\_tbnimatrix, 95
  - show\_tbniscr, 96
  - show\_tbniscrall, 97
  - show\_tbniscrplotly, 98
  - show\_tdlcrk, 99
  - show\_tdlcrkindic, 99
  - show\_tdlcrkindiccdf, 100
  - show\_tdlcrkline, 101
  - show\_tdlcrkmatrix, 102
  - show\_tdlcrkradar, 103
  - show\_thrplot, 104
  - show\_transect, 105
  - show\_transectavespp, 106
  - show\_transectmatrix, 108
  - show\_transectsum, 110
  - show\_wqmatrix, 111
- \* util**
  - util\_html, 125
  - util\_map, 126
  - util\_ordin, 127
- acres, 4
- addPolylines, 99
- anlz\_attain, 5, 6
- anlz\_attainsite, 6, 92, 93
- anlz\_avedat, 5, 7
- anlz\_avedatsite, 6, 7
- anlz\_enteromap, 8, 70
- anlz\_fibmap, 9, 9, 71
- anlz\_fibmatrix, 11, 73
- anlz\_fibwetdry, 8, 9, 11, 13, 70, 73, 75
- anlz\_hmpreport, 14
- anlz\_hydroload, 16
- anlz\_iwrraw, 17
- anlz\_refs, 17
- anlz\_sedimentaddtot, 18
- anlz\_sedimentave, 19
- anlz\_sedimentpel, 18, 20
- anlz\_sedimentpelave, 21
- anlz\_tbbimed, 22
- anlz\_tbbiscr, 22, 23, 94
- anlz\_tbniave, 23
- anlz\_tbnimet, 24, 25
- anlz\_tbniscr, 24, 25, 96–98
- anlz\_tdlcrk, 26, 99, 102

- anlz\_tdlcrkindic, [26](#), [100](#), [103](#)
- anlz\_transectave, [27](#)
- anlz\_transectavespp, [28](#)
- anlz\_transectocc, [28](#), [29](#), [30](#), [107](#), [109](#), [110](#)
- anlz\_yrattain, [31](#)
  
- benthicdata, [31](#)
- bsmap, [32](#)
  
- catchpixels, [32](#)
- catchprecip, [8](#), [12](#), [13](#), [33](#), [51](#), [70](#), [73](#), [75](#)
  
- data.frame, [17](#), [26](#), [27](#)
  
- enterodata, [8](#), [34](#), [70](#)
- epcdata, [35](#), [81](#)
  
- fibdata, [36](#)
- fimdata, [37](#)
- fimstations, [38](#)
- flextable, [67](#), [81](#)
- fromJSON, [47](#)
  
- ggmap, [32](#)
- ggplot, [68](#), [69](#), [76](#), [79](#), [84](#), [86](#), [88](#), [91](#), [95–98](#), [102](#), [105–107](#), [109](#), [112](#)
- ggplot2, [78](#)
  
- hmptrgs, [39](#)
- HTML, [126](#)
- <https://data-sfwfmd.opendata.arcgis.com/>, [83](#)
  
- iconList, [125](#)
- iwrraw, [39](#)
  
- leaflet, [87](#), [89](#), [99](#)
  
- phytodata, [40](#)
- plotly, [76](#), [80](#), [84](#), [86](#), [88](#), [95](#), [97](#), [98](#), [107](#), [109](#), [110](#)
- position\_jitter, [68](#)
  
- reactable, [75](#), [76](#), [79](#), [82](#), [107](#), [112](#)
- read\_dlcurrent, [41](#), [53–56](#), [59](#)
- read\_formbenthic, [42](#)
- read\_formfib, [43](#), [54](#)
- read\_formfim, [44](#)
- read\_formphyto, [45](#)
- read\_formsediment, [46](#)
- read\_formtransect, [47](#)
  
- read\_formwq, [48](#), [55](#), [59](#)
- read\_formwqp, [49](#), [60](#)
- read\_importbenthic, [23](#), [42](#), [50](#)
- read\_importentero, [9](#), [11](#), [12](#), [51](#), [72–75](#)
- read\_importtepc, [43](#), [48](#), [52](#)
- read\_importfib, [10–12](#), [43](#), [53](#), [53](#), [54](#), [71–74](#), [76](#)
- read\_importfim, [25](#), [44](#), [55](#)
- read\_importphyto, [45](#), [55](#), [56](#), [59](#)
- read\_importtrain, [8](#), [12](#), [13](#), [57](#), [70](#), [73](#), [75](#)
- read\_importsediment, [18–21](#), [46](#), [57](#), [84–87](#), [89](#)
- read\_importwq, [31](#), [48](#), [53](#), [56](#), [58](#), [67](#), [68](#), [78](#), [80](#), [90–93](#), [104](#), [111](#)
- read\_importwqp, [49](#), [60](#), [127](#)
- read\_transect, [30](#), [61](#), [69](#), [105](#)
- read\_xlsx, [53](#), [54](#), [56](#), [59](#)
  
- scale\_x\_discrete, [95](#), [96](#), [109](#)
- seagrass, [62](#), [83](#)
- sedimentdata, [63](#)
- set\_caption, [67](#)
- sf, [17](#), [26](#), [38](#), [64](#), [65](#), [113](#), [117–120](#)
- sgmanagement, [64](#)
- sgseg, [65](#)
- show\_annualassess, [66](#)
- show\_boxplot, [67](#)
- show\_complot, [68](#)
- show\_enteromap, [8](#), [9](#), [69](#), [125](#)
- show\_fibmap, [10](#), [71](#), [125](#)
- show\_fibmatmap, [72](#), [125](#)
- show\_fibmatrix, [12](#), [13](#), [74](#)
- show\_hmpreport, [76](#)
- show\_matrix, [67](#), [78](#), [80](#), [81](#), [91](#), [112](#)
- show\_matrixplotly, [79](#)
- show\_ratab, [80](#)
- show\_reactable, [81](#)
- show\_seagrasscoverage, [82](#)
- show\_sedimentalratio, [83](#)
- show\_sedimentave, [20](#), [85](#)
- show\_sedimentmap, [86](#)
- show\_sedimentpelave, [22](#), [87](#), [88](#)
- show\_sedimentpelaveplotly, [88](#)
- show\_sedimentpelmap, [21](#), [89](#)
- show\_segmatrix, [79](#), [80](#), [90](#), [92](#), [112](#)
- show\_segplotly, [91](#)
- show\_sitemap, [92](#), [94](#)
- show\_sitesegmap, [93](#)
- show\_tbbimatrix, [94](#)

show\_tbnimatrix, [80](#), [95](#)  
show\_tbnisr, [96](#), [98](#)  
show\_tbniscrall, [97](#), [98](#)  
show\_tbniscrplotly, [98](#)  
show\_tdlcrk, [99](#)  
show\_tdlcrkindic, [99](#), [101](#)  
show\_tdlcrkindiccdf, [100](#), [101](#)  
show\_tdlcrkline, [101](#)  
show\_tdlcrkmatrix, [102](#)  
show\_tdlcrkradar, [26](#), [27](#), [103](#)  
show\_thrplot, [92](#), [104](#)  
show\_transect, [105](#)  
show\_transectavespp, [106](#)  
show\_transectmatrix, [108](#)  
show\_transectsum, [110](#)  
show\_wqmatrix, [79–81](#), [91](#), [111](#)  
stations, [112](#)  
subtacres, [113](#)  
swfwmdtbseg, [113](#)

targets, [5](#), [6](#), [68](#), [79](#), [90](#), [92](#), [93](#), [104](#), [111](#), [114](#)  
tbniref, [115](#)  
tbnispp, [116](#)  
tbseg, [64](#), [117](#), [118](#)  
tbsegdetail, [118](#)  
tbseglines, [118](#)  
tbsegshed, [119](#)  
tbshed, [119](#)  
tibble, [12](#), [23](#), [31](#), [40](#), [42](#)  
tidalcreeks, [17](#), [120](#)  
tidaltargets, [26](#), [121](#)  
transect, [122](#)  
trnlns, [123](#)  
trnpts, [123](#)

util\_fibicons, [124](#)  
util\_fiblevs, [125](#)  
util\_html, [125](#)  
util\_map, [126](#)  
util\_organ, [127](#)